Optimization of a Data-Driven Customer Relationship Management (CRM) System for Better Decision Making

Confidential

Master thesis performed in collaboration with ING Private Banking and submitted to Delft University of Technology in partial fulfilment of the requirements for the degree of
Master of Science
in **Engineering and Policy Analysis**

Faculty of Technology, Policy and Management

by

Milad Khajehvajari

Student number: 4388976

To be defended in public on: 06/08/2019

Graduation committee

Chair/First Supervisor: Dr. Scott Cunningham, Policy Analysis of Multi-Actor Systems

Second Supervisor: Dr. Yilin Huang, Systems Engineering & Simulation

External Supervisor: Damir Fific, Lead Orchestration, ING Private Banking

Acknowledgements

This project was performed in collaboration with the ING Private Banking and OneAnalytics departments, as well as the TPM Faculty of TU Delft.

I’d like to thank Jens and Damir from the OneAnalytics team for giving me the opportunity to perform this project at the Private Banking department. Furthermore, thanks to Damir (again!) and Nguyen for their extremely valuable assistance in the different stages of the project, especially at places when I was really stuck and asked an un-ending barrage of questions (yes, I know!). The past 5 months have been very interesting, fairly challenging and in the end, extremely rewarding.

I’d also like to thank Dr. Scott Cunningham, the chair of this thesis project with whom I’ve performed two internships in the past year, including this one. It certainly hasn’t always been easy, but it’s been a pleasure. Thank you for making sure always the scope of the project is aligned correctly and going in the right direction. Furthermore, I’d like to thank Dr. Yilin Huang, the 2nd supervisor of this thesis project for providing very sharp and very helpful feedback.

Finally, I’d like to thank my parents and family for the immeasurable sacrifices that they’ve made in the past 8 years for me to be able to live in and gain education across 3 different countries. Without your help none of this would have been possible and although I perfectly understand the reasons behind it, it really is a great shame that you cannot make it to the graduation ceremony.

Table of Contents

Acknowledgements 1

Table of Contents 2

List of Figures 4

List of Tables 6

1 Introduction & Problem Context 7

1.1 Introduction to the Lead Management System 9

1.2 Research Question(s) 14

2 Theory 16

2.1 On Analytics in CRM 16

2.2 On GDPR Compliant Data Processing 19

3 Methodology 26

3.1 Algorithms 26

3.1.1 Logistic Regression 27

3.1.2 Random Forests 28

3.1.3 Multi-Layer Perceptron 30

3.2 Algorithm Metrics 32

3.2.1 Precision, Recall, F1 Score and Accuracy 32

3.2.2 k-Fold Cross-Validation 33

3.3 Hyperparameter Tuning 33

3.3.1 Grid Search & Random Search 34

3.4 Banker Preference Survey 35

4 Exploratory Data Analysis & Modeling 36

4.1 Data Exploration 36

4.2 Model Generation & Results 40

4.2.1 Addition of Team Data 42

4.2.2 Addition of Customer Potential 44

4.3 Model Parameter Tuning & Results 46

4.4 Model Validation & Best Model Selection 48

5 Testing & Results 50

5.1 Urgent & Non-Urgent Leads 50

5.2 Exploring the Random Forest 51

5.2.1 Priors 58
5.3 A/B Testing 59
5.3.1 A/B Test Setting 59
5.3.2 Results 61
5.3.3 Discussion and Evaluation 65
5.4 Banker Preference Survey Results 72
6 Overview & Conclusion 77
6.1 Project Overview 77
6.2 Conclusion & Future Steps 80
6.3 Personal Reflection 84
References 87
Appendix I: Additional Tables 92

List of Figures

Figure 1: Representation of a typical CRM system. Reprinted from "Theoretical Models of Customer Relationship Management in Organizations" by Tavana et. al - International Journal of Business and Behavioral Sciences, 2013........8
Figure 2: Systematic overview of the current lead management system 12
Figure 3: Layout of this report 15
Figure 4: Lead management areas of relevance. Reprinted from ‘Strategic Approach to Optimizing Leads Management Process’ by Ripal Patel - Capgemini , 2015........17
Figure 5: Systematic overview of the proposed lead management system........18
Figure 6: Prediction Accuracy vs. Explainability. Reprinted from “Explainable Artificial Intelligence (XAI)” by David Gunning - Defense Advanced Research Projects Agency (DARPA), 2017........23
Figure 7: Representation of a typical decision tree........28
Figure 8: A representation of a one (hidden) layer MLP 30
Figure 9: Visual representation of precision and recall 32
Figure 10: Frequency of each label in the training data........36
Figure 11: Frequency of triggers in certain leads........37
Figure 12: Frequent triggers with the highest conversion rates 38
Figure 13: Pairplot between age and outcome of contact........38
Figure 14: Example code to create, train and save model object 40
Figure 15: Example code to load model, match feature matrix and make new predictions 41
Figure 16: Runtime and performance of the three algorithms with the initial feature matrix 41
Figure 17: Runtime and performance of the three algorithms with the compact feature matrix..42
Figure 18: Aggregate team performance per region........43
Figure 19: Aggregate performance of five most active regions across twenty highest performing triggers........43
Figure 20: Runtime and performance of the three algorithms after addition of team ids 44
Figure 21: Distribution of customer potential across successful (1) and unsuccessful (0) leads 45
Figure 22: Runtime and performance of the three algorithms after addition of team ids........45
Figure 23: Example of a random search for with a random forest model........46
Figure 24: Performance of the three algorithms after parameter tuning........47
Figure 25: Performance of the three models with new data from the test week........49
Figure 26: Distribution of the percentage of pure leaf nodes over the trees of the forest........51
Figure 27: Distribution of the total number of nodes over the trees of the forest........52
Figure 28: Start and end of the decision path of two trees of the forest........53
Figure 29: Overall feature importance of the model........54
Figure 30: Six highest marginal contributions to the final probability calculation for the lead 55
Figure 31: Distribution of tree depth over the forest........56
Figure 32: Distribution of pure leaf nodes (left) and number of nodes (right) of the trees over the forest........56
Figure 33: Overall feature importance of the modified model 57
Figure 34: Five highest marginal contributions to the final probability calculation for the lead....58
Figure 35: Systematic overview of the A/B testing........60

Figure 36: Operational overview of the A/B testing..61
Figure 37: Urgent vs. non-urgent leads conversion 62
Figure 38: Urgent vs. non-urgent lead engagement 62
Figure 39: Random forest vs. existing method model performance 63
Figure 40: Random forest vs. existing method outcome distribution..64
Figure 41: Random forest lead conversions at different probability thresholds 65
Figure 42: Top 10 most selected triggers for the two models across 6 weeks of testing............66
Figure 43: Top 10 most common triggers in successful leads per model over the 6 weeks.......68
Figure 44: Feature importance of leads from week 4 (top row) and week 5 (bottom row)..........69
Figure 45: Calculated probability of success over the leads sent out by the random forest model 70
Figure 46: Calculated probability of success of urgent & non-urgent leads over 6 weeks..........71
Figure 47: Trigger preferences of the seven participating regions 73
Figure 48: Performance vs preference of the participating regions over the ten triggers 75
Figure 49: Potential areas for future research 83

List of Tables

Table 1: Trigger categories and their encoding 11
Table 2: Overview of the three Algorithms..31
Table 3: Features of the selected lead 54
Table 4: Features of the selected lead 57
Table 5: Triggers for each of the selected leads in weeks 4 & 5...69
Table 6: The ten triggers selected for the survey..72
Table 7: Example boosting figures for implementation of preferences into the scoring method .76
Table A-1: Accuracy score & runtime of algorithms per fold in the initial setting 92
Table A-2: Accuracy score & runtime of algorithms per fold with the compact feature matrix 93
Table A-3: Accuracy score & runtime of algorithms per fold after the addition of team ids.........94
Table A-4: Accuracy score & runtime of algorithms per fold after the addition of customer potential 95
Table A-5: Hyperparameters per algorithm 96
Table A-6: Accuracy score & runtime of algorithms per fold after hyperparameter tuning..........97
Table A-7: List of all urgent triggers...98

1 Introduction & Problem Context

ING is a global financial institution based in the Netherlands, serving more than 38 million customers and having more than 50000 employees worldwide (ING, 2019). As financial markets have evolved and become more digital, the ways of marketing, communication and customer service have also adapted to the times. Customers are gaining more experience with better service in other industries and thus expecting a higher standard from financial services as well (PwC, 2016). Research has shown that customer experience will become the key brand differentiator by 2020 (Walker, 2013). This makes the acquisition and retention of customers ever more important in all industries, financial services included.

On top of being a nuisance to unsatisfied customers, poor customer service costs industries globally ＄338.5 billion in potential revenue losses per year. The industry with the highest losses is financial services, with about ＄44 billion lost per year (Genesys, 2009). In the Netherlands, the banking sector has a relatively large asset proportion to the GDP, which has decreased from 530% in 2007 to 370% in 2017. (European Banking Federation, 2018). Nevertheless, with increasing utilisation rates, the demand for external credit is expected to increase. Considering that banks play a vital role in the allocation of capital in the form of different types of funding, it is then of utmost importance that the decisions regarding this allocation are made in an informed manner. In recent years, the profitability of Dutch banks have somewhat recovered, but issues such as the low interest rates, increased regulatory burdens and the need for investments in innovation and digitalization remain challenging (European Banking Federation, 2018).

After the 2007 crisis, financial institutes have been investing heavily on their outdated IT infrastructure (PwC, 2016). Part of the IT infrastructure is used for the collection and storage of customer data through multiple channels. It is natural then that this information may be used to improve the services of the financial institutions in some way. Systems utilizing customer data for the purpose of improving business relationships with customers through better customer communication and service are called Customer Relationship Management (CRM) models (Chen & Popovic, 2003).

The concept of this project is related to the optimization of a data-driven CRM system in the context of private banking. The system is used to support decision making of the private bankers making calls to customers across the Netherlands. The project entails intertwined technical and societal components. The multidisciplinary nature of the project in using quantitative techniques on top of qualitative theory in order to support decision making and benefit both the corporation and in turn the society aligns the project well with the standards of the EPA program. The general layout of a CRM can be seen in Figure 1 below.

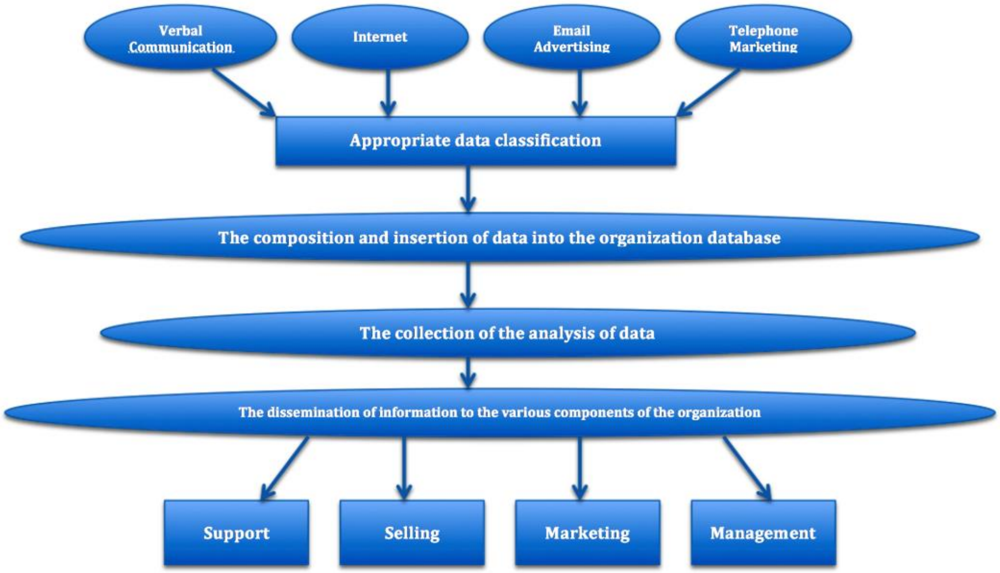

Verbal Communication
Internet
Email Advertising
Telephone Marketing
Appropriate data classification
The composition and insertion of data into the organization database
The collection of the analysis of data
The dissemination of information to the various components of the organization
Support
Selling
Marketing
Management

Figure 1: Representation of a typical CRM system. Reprinted from "Theoretical Models of Customer Relationship Management in Organizations" by Tavana et. al - International Journal of Business and Behavioral Sciences, 2013

Highly successful customer relations translates to more efficient service and happier customers, while maintaining potential for revenue for the company. This is usually measured by calculating the fraction of customer contacts made based on the models recommendations which meet some form of predefined success criteria. This could be making a sale, making an appointment for discussing services, etc. This fraction of successful contracts can then be called the success rate. In this project, a higher success rate is to be achieved by optimizing a lead management system, which is a type of CRM. Currently, the model in place has a success rate of only about 50%, which is already considered above average by the company based on their experience. Based on customers activities certain “triggers” are generated and stored in a database. For example, a customer may be interested in a mortgage, or in opening bank accounts for their children, etc.

These triggers are then ranked based on historical data, the most promising ones are bundled into so-called “leads”. While a large number of leads may be generated, the capacity for contact is not large enough for all leads to be contacted. Therefore, from the pool of all leads, the n best leads must be selected for distribution (n being the capacity for contact). Naturally, to make this selection, some method of scoring the leads must be implemented. This is a main area of research for this project, where the potential for implementing a data analytics approach for the scoring of the leads is to be investigated.

The next step is the distribution of these leads (dissemination in Figure 1). Currently this is mainly done by sending the leads to private bankers, which then may or may not contact customers regarding those leads. Here, there is another area of focus for the research, where the way the leads are distributed to private bankers can be made smarter based on their preference (or expertise) and utilization. For example, a private banker with 10 years expertise in mortgages may have a preference for and thus be more likely to contact a customer for mortgage discussions, compared to a private banker with no experience in mortgages at all.

Additional to the technical requirements of the project, legal compliance with regards to data practices is an important consideration. Models such as ones used in this project track, generate, store and process data about customers algorithmically. In this way, customers are seen as valuable walking data generators, while not necessarily being aware of how this data is processed in the algorithms which in turn lead to decision-making. The newly passed General Data Protection Regulation (GDPR) in the European Union affects these kind of activities directly. As a result, it is important to establish how the practice stands with regards to this regulation and compliance. Part of compliance concerns the governance of decision making based on machine learning algorithms such as the ones used in this project. Therefore, it must be investigated whether additional measures need to be taken in order to enable better governance of the algorithmic decision making process and/or to ensure general compliance with the regulation.

In the following section, an introduction of lead management in general and the system which is used in this project in particular is provided. The current way in which the lead management system scores the leads is provided and the case for using analytics for the scoring is made. Additionally, the prospect of considering preference-based routing for the distribution of the leads is investigated.

1.1 Introduction to the Lead Management System

Lead management in particular is the set of methodologies, systems and practices with the aim of helping to better service existing clients in a more informed manner (retention) or discovering potentially new clients (acquisition). Generating and improving the quality of leads is often reported as one of the top challenges in the marketing industry (Krogue, 2015). Despite advances towards data-based approaches, most of lead generation is still based on guesswork and the business knowledge of managers who believe the know the key characteristics which contribute to successful leads in their respective businesses (Monat, 2011).

Lead generation in the private banking department at ING was also done in a manual way with bankers choosing certain criteria about customers which they wanted to contact based on own judgement and on an ad-hoc basis. Based on that choice, from the data analytics department a list of customers was sent, which then may or may not have been contacted. There was no feedback loop from the bankers, so there is no data available with regards to conversion or even engagement rates from the time. In order to improve this manual process, it was chosen to replace it with a data-driven approach, which is the current system in place.

The current system in place for private banking which generates triggers, creates leads from those triggers, sends out the chosen leads and receives and stores the feedback is called REVA (implemented fully from 2017). As mentioned earlier, leads are sent out either for customers who are already users of private banking (retention), or for indicating potential customers who could become users of private banking (acquisition). The pool of all possible triggers has been manufactured by experienced staff at the ING private banking department based on potential indicators of success in previous calls with customers. Everyday, the whole customer database is searched and every customer is assigned all the triggers for which they meet the underlying criteria (this is checked based on the available data).

Note that, this means for some triggers they will be repeated for many consecutive days, because the customers condition has not changed. Each trigger is generated as a result of a query on the customer database, with the output being an aggregation of data received from multiple sources such as the ING website (activity, traced by cookies), ING branches (mostly for visits), ING add-ins in certain websites (e.g mortgage calculator, etc), other ING departments (not private banking, mostly for identifying prospective customers) and sometimes external sources such as the KvK, etc. The generated triggers can be categorized by the general purpose that they serve. These purposes are explained below:

1- Financial Capital Availability: are either certain conditions (simply yes/no) or proxy indicators of availability of capital from the customer, which in turn means possibility to offer investment/wealth management services. This data is either available directly to ING or sometimes is gathered from external sources. These will be encoded by the letter ‘F’ when being referred to later in figures and/or text.

2- Compliance: about compliance of service with a timeframe, namely revision of customer investment portfolio/loan conditions and are in place to ensure legal compliance. These will be encoded by the letter ‘C’ when being referred to later in figures and/or text.

3- Customer Contact/Multiple Services: proxy indicators which in specific cases could translate to interest in services like opening bank accounts for children/younger family members, adding a 2nd person (giving management rights) on an elderly person’s bank account, mortgages, etc. Additionally, triggers indicating a customer has not been contacted for a long period of time are also categorized under this. These will be encoded by the letter ‘M’ when being referred to later in figures and/or text.

4- Customer Opinion/Feedback: in order to track customer opinion of services offered, in order to reinforce positive opinion or try and change negative/neutral opinion. These will be encoded by the letter ‘O’ when being referred to later in figures and/or text.

5- Data Integrity: These track possible mismatches or temporal inaccuracies in customer data, from a changed house address to a mismatch in identification, it is important both for information accuracy and often legal compliance. These will be encoded by the letter ‘I’ when being referred to later in figures and/or text.

6- Product Ownership and Services: about ownership/expiration of or interest in certain products or services. An indication of the need for re-negotiation of services and/or a change in the products offered to customers. These will be encoded by the letter ‘P’ when being referred to later in figures and/or text. A summary of the trigger categories and their corresponding encoding can be found in Table 1 below.

Table 1: Trigger categories and their encoding

Category	**Encoding**
Financial Capital Availability	F
Compliance	C
Customer Contact/Multiple Services	M
Customer Opinion/Feedback	O
Data Integrity	I
Product Ownership and Services	P

For the purposes of this model, it is assumed the trigger generation mechanism is behaving perfectly, that is to say these conditions are equally applied to all customers who meet the underlying criteria, and therefore there is no bias in the input data (or at least, there is no reason to believe there is). The top five triggers are selected based on the historical success rate of the leads that they have been present in. That is:

$$1.1.1 \quad Success(T) = \frac{N_s}{N}$$

Where T is any given trigger, N_s is the number of successful leads T has been a part of and N is the total number of leads which T has been a part of. Following this calculation, the top 5 triggers with the highest success are then selected for the creation of the lead. A simplified representation of the REVA system can be found in Figure 2 below.

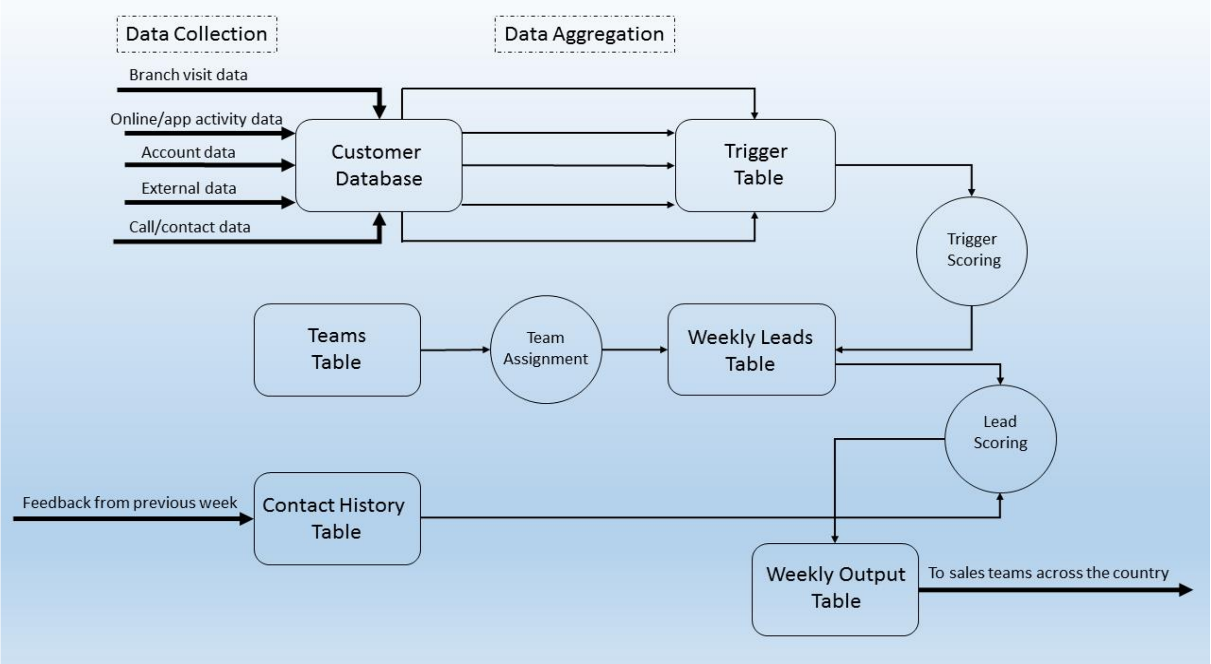


Figure 2: Systematic overview of the current lead management system

Going from top left to bottom right, the customer data is first queried from multiple tables and collected in one trigger table based on whether the conditions of each trigger are met in the data. Then, these triggers are each scored according to equation 1.1.1 explained above. The leads table is created by assigning a score to each lead, which is the sum of the scores of the triggers which make up the lead, as seen below in equation 1.1.4:

$$1.1.2 \quad Score(T_1) = Success(T_1) * 100$$

$$1.1.3 \quad Score(T_n) = \frac{Success(T_n)}{Success(T_1)} * 100$$

$$1.1.4 \quad Score(lead) = \sum_{n=1}^{5} Score(T_n)$$

Where T_1 is the trigger with the highest historical conversion. Therefore, the historical success of each other trigger in the lead is adjusted by that of the highest performing trigger and eventually all the scores are summed to arrive at the final score for the lead. Additionally, each lead assigned to a team depending on geographical proximity of the team to the corresponding customer. After the scoring and team assignment, the required number of leads with the highest scores per team are sent out. Note that in this way, the selection criteria of leads changed from the judgement of the bankers to the performance of the contact indicators, hence the data-driven approach. Another additional benefit was the addition of the feedback loop, based on which the triggers and subsequently the leads are scored each week. In addition, the process is much more consistent, with a certain number of leads being sent out every week, rather than whenever the bankers asked for them.

That is not to say the current process is without its limitations. The lower threshold for the frequency triggers to be considered for scoring in this way is set at eight and as a result, it is possible that triggers with very few appearances get disproportionately high scores (noting that the total data points over which the scoring is done is over 20000 rows). On the other hand, the frequently appearing triggers which already have a high score would keep being selected week after week. Therefore, it is also prone to slow adaptation in case a usually good trigger performs badly for a few weeks. Furthermore, the only metric of trigger selection is a count on the success of the triggers, and the nature of the trigger, the customer and the team which will handle the lead are not considered whatsoever.

For the scoring of the leads, the hypothesis is that by learning from the previous data using an analytical process, patterns can be better identified in order to have a more informed prioritization than the current method. Such methods have been used previously in other sections of marketing which have similar purposes. Some examples are different techniques in times series analysis (Ji, Wang & Zhang, 2016), logistic regression (Shao & Li, 2011), bayesian statistics (Rutz & Bucklin, 2011), markov models (Abhishek, Fader & Hosanagar, 2012) & game theory using the Shapley value (Berman, 2013; Yadagiri, Saini & Sinha, 2015). Consequently, based on previously established theory, using data modelling could improve the accuracy of the lead selection.

This is due to the more thorough evaluation of the data and the fact that more information can be out into the algorithm. The technical knowledge gap in this project then, is the potential integration of data analytics techniques using machine learning algorithms as part of the lead management system and the exploration of their usefulness in the scoring of the leads, in such a way to increase the overall success rate.

However, it is worth noting the starting point for a data-based predictive model is far from ideal. There are two limitations to the input information of the model. One is that the leads available as learning data for the model are only the ones that have been delivered by the previous system and hence have received feedback. Additionally, the top five triggers which are available for each lead are also selected based on the previous scoring and as a result, any information from other triggers for each customer which were available at the time is lost. Given the current circumstances, it is not possible to mitigate either of these issues, as the data in either case was not saved. It is nevertheless important to recognize these issues and perhaps take steps to change them for future iterations of any potential models.

Other than the scoring of the leads, the distribution step assigns this information to a channel, through which customers are contacted. In the case of private banking, this channel is usually a private banker which may contact the customer via a phone call. Some leads may not necessarily be bad, but the customer contact point may not do a good job of presenting the lead (closing the sale). Therefore, data about the preference (or competency) and utilization of sellers can be used to make a smarter distribution, thus mitigating (or at least considering) the negative effects of the contact point on the process. Data about which team every lead has been sent to is available, and a survey will be conducted as well to gain an insight into the preferences of the teams with regards to the type of leads they receive. Using this data, opportunities for optimizing the distribution process will be investigated. It is worth noting that there will be some limitations to this.

The distribution of the leads is done on a team level, meaning a contact team receives the lead rather than an individual banker. There are one or more teams per region in the Netherlands. Previous research by the company has shown that private banking customers have a strong preference for being served only in the region in which they live. Therefore, before any other considerations, a lead from a certain region can only be sent to teams in that region. Having established the information presented so far, in the following section, the main research question and corresponding sub-questions are defined.

1.2 Research Question(s)

Overall, summing up both the technical and societal part of the project, the main research question arrived at is as follows: '*How can a data-driven approach using machine learning be used in the lead management system in order to increase the success rate, while considering and applying appropriate measures to ensure compliance with data protection laws such as the GDPR?*'

The relevant subquestions (SQs), which are defined based on the information presented previously in this chapter are formulated below.

SQ1- What are the necessary measures for compliance to be ensured in the design of algorithms for data-driven lead management in light of the GDPR regulations?

SQ2- How can a data analytics driven scoring method be integrated into the existing lead management infrastructure at the bank?

SQ3- Based on a data driven approach, which n number of leads (n being the corresponding capacity for customer contact) have the highest probability of success (making a sale) at any particular time?

SQ4- What is the effect of the preferences of the channels of contact (private bankers) on the success of the process and how can the lead distribution be optimized in this regard?

Having laid out the research question and sub-questions, next a layout for the research and this report is presented. This can be seen in Figure 3 below:

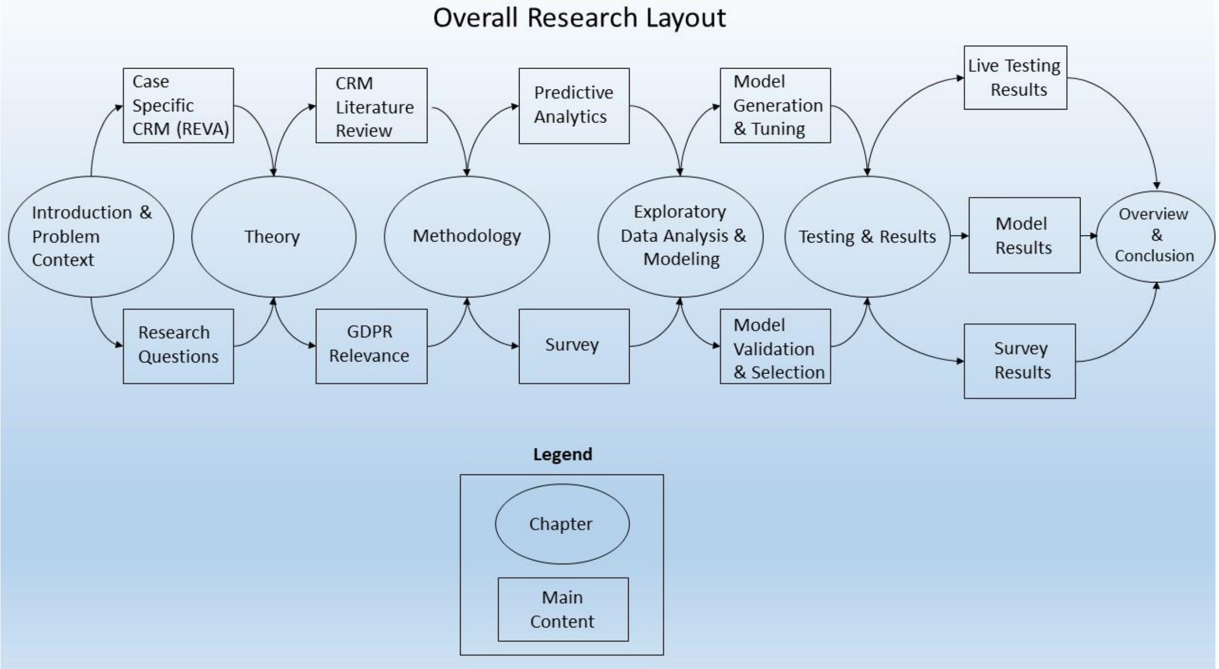


Figure 3: Layout of this report

Following from this section, in the theory chapter further review of relevant literature is presented which mostly relate to SQ1 and SQ2. In the Methodology chapter, based on the theory presented in the previous chapter, the relevant research methodology for the data analytics is presented and the technical terms are explained briefly. Additionally, the design of the survey to be conducted with the bankers is presented. In the following chapter, the execution of this methodology is presented as well as analysis of the data, which results in the creation, tuning and validation testing of three separate algorithms.

Based on the result of the validation testing, the model with the best performing algorithm then moves on to the Testing and Results chapter. In this chapter, the potential integration of the data model into the existing lead management infrastructure is investigated and explained. Additionally, the details and setting of the live testing against the existing scoring method is presented. Furthermore, the structure of the model is explored and the results of the testing are presented and discussed, which relates mostly to SQ3. Insights can be gained into whether the data analytics approach has improved results and if so, on what basis.

Additionally, the results of the survey as described in the methodology are also presented and discussed, which relates mostly to SQ4. Insights can be gained into whether in the case of this project, the preferences of the channels of contact have any effect on potential performance and if so, how can this be integrated into the lead selection process. In the final chapter, based on all the previous research and results, the main research question and SQs as defined in this section are answered. Next, the main results and limitations of the research are discussed, and recommendations for both the existing system and potential future research are presented. Finally, an informal personal reflection is presented.

2 Theory

In this chapter the theory upon which the following chapters are built on is explained. First, a review of the relevant CRM literature is provided and the potential use of data analytics is explained. This is followed by an introduction of the GDPR and its relevance to this project in particular. This section includes the presentation by the measures taken by ING to ensure compliance with the GDPR, as well as to prevent disproportionate harm to data agents. Building on the existing measures and discussions of explainability, it is then proposed to add an extra dimension to these measures based on explainability for better interpretation of model decisions.

2.1 On Analytics in CRM

Customer behaviour is a complex topic which can often deviate from the rational decision making principles assumed to be true in traditional economics, as is extensively argued by behavioural economics (Lin, 2012). It has been shown that people tend to have knowledge-actions gaps (Kennedy et al., 2004) , value-action gaps (Flynn et al., 2009), attitude-action gaps or intention-action gaps in their decision-making (Boulstridge & Carrigan, 2000; Sheeran, 2002). All of this leads to an imperfect rationale in decision making. To identify patterns and gain insights with regards to behaviour, the collection, storage and analysis of data specific to the industry and business can then be helpful. This is where customer relationship management can add value. Aside from the systematic definition provided earlier from Chen & Popovic (2003), CRM has been defined to have evolved from advances in information technology as well as organizational changes which are becoming more focused on the customers (Parvatiyar & Sheth, 2001). There are different categorizations of CRM available, here a high-level distinction based on functionality is used (Fayerman, 2002):

1- Operational CRM: involves the automation of business processes regarding the front-office of customer contact. Typical functions include sales automation, marketing automation, customer service automation, etc. Lead management is an activity regarding marketing automation and as such falls under this category of CRM.

2- Analytical CRM: involves the full cycle of data operations ranging including collection, storing and processing, among others. Often integrated into operational CRM.

3- Collaborative CRM: involves making infrastructure available for use of collaboration between different channels and/or departments within the company for CRM activities such as data collection, among others.

The way the lead management will be performed in this project is an integration of all three categories of CRM, driven by customer data. The lead generation is based on a collection certain information (triggers) which are gathered based on data about customers which is collected through different channels (for example social media, online activity, in-branch visits, etc). This is the use of collaborative CRM. Based on the analysis of this data, some form of selection will be made between the leads, which is the use of analytical CRM. Finally, based on the results of the analytical CRM a certain number of leads are automatically sent out to sales teams across the country, which is the use of operational CRM. A graphical representation of a typical lead management system with the relevant areas can be seen in Figure 4 below.

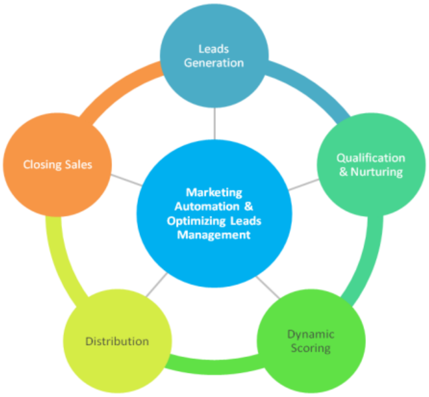

Leads Generation
Qualification & Nurturing
Dynamic Scoring
Distribution
Closing Sales
Marketing Automation & Optimizing Leads Management

Figure 4: Lead management areas of relevance. Reprinted from ‘Strategic Approach to Optimizing Leads Management Process’ by Ripal Patel - Capgemini , 2015

For the optimization of lead selection, the focus will be on the dynamic scoring component. This entails ranking the different leads by likeliness of being successful (and their corresponding customers) in the database based on some form of analysis in order to select the best leads (that is the ones which are most likely to be successful) for distribution. In this specific instance of lead management, aggregated data is scored using a count (for details, please refer to section 1.1) in order to select the best leads. The use of data modelling has not been tested so far with this specific system, and its integration as part of the system is part of the main goals of this research.

Part of analytical CRM defined earlier concerns the analysis of customer data in order to gain insights with regards to characteristics and behaviours with the aim of supporting the organization’s customer management strategies (Teo et al., 2006). One means of analyzing any form of data (customer data included) are data mining & analytics tools. For a successful CRM implementation, two important aspects are required (K. Tsiptsis, 2018):

1- Appropriate IT infrastructure to store and process customer data.

2- Analytical processes in order to transform this data into valuable customer insights.

There is already an infrastructure in place for the storage of the data, which is used by the current scoring method. Multiple machine learning algorithms can be used in order to learn from customer data collected by the business in order to improve customer retention and acquisition. In order to do so, integration into the existing infrastructure must be considered. Using a similar structure to the one presented in Figure 2 in section 1.1, the following systematic structure is proposed, as seen in Figure 5 below:

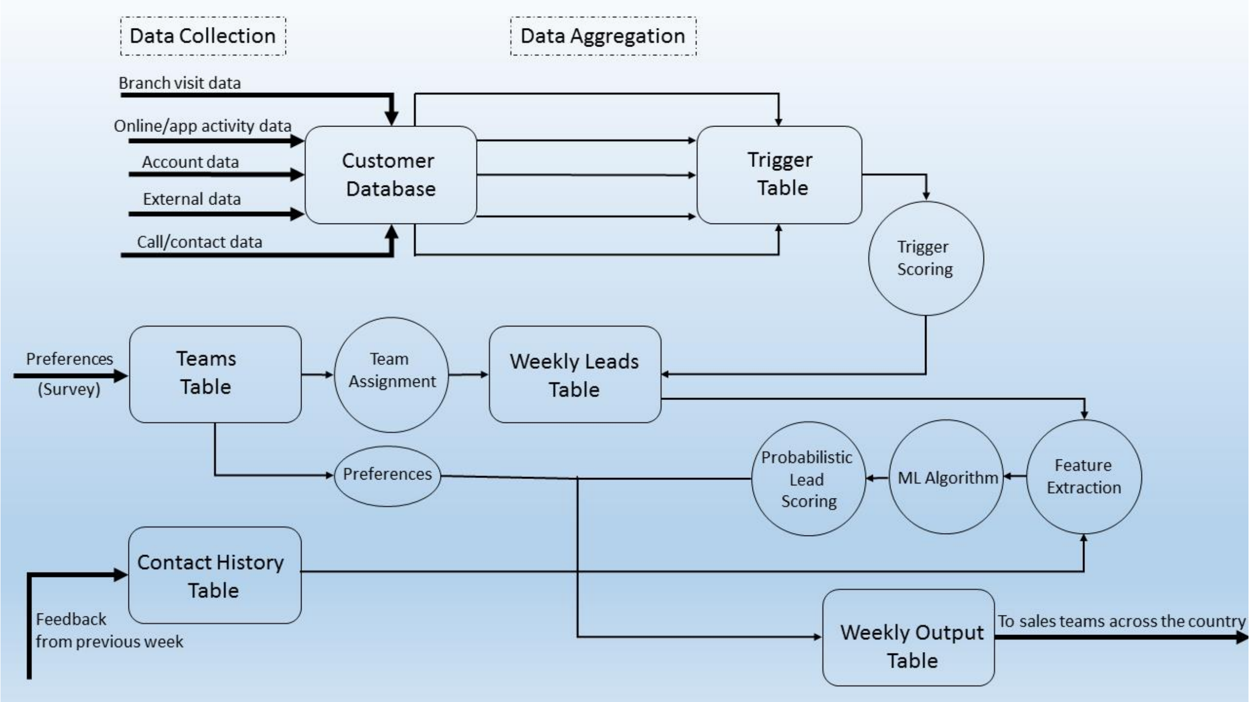


Figure 5: Systematic overview of the proposed lead management system

Up to the creation of the weekly leads table, the process is exactly the same as the existing method, which was explained in the previous chapter. However, rather than the traditional scoring method, the data in the leads can be extracted as features (which in this case would be binary factors, as most of the data is categorical) and fed to a machine learning algorithm in order to calculate a probability of success for the lead. On top of that, by performing a survey from the bankers who make the calls, their preferences can be used to modify these probabilities, based on the assumption if they receive more of their preferred leads, they are less likely to pass up on a lead. The exact tools and manner of integration of the data analytics approach into the infrastructure will be presented and explained in chapter 5 (see section 5.3.1).

Additionally, each week new data will become available in the form of the feedback received from the previous weeks, which can then again be fed into the model for further learning. This way, the model is learning in batch form every week, with the batch incrementally increasing in size per week based on the feedback of the previous week (batch refers to the fact that the model is trained on all data points at once). Then, it is expected that a reinforcing loop is created between the model and the bankers. The model adapts every week based on the feedback of the bankers from the previous week, and then makes decisions on which leads to select for the upcoming week, which are then sent out to the bankers who are making their decision based on these results.

Another topic to consider is the preferences of the bankers. Previous research has shown that by incorporating call center agents more in the process of selecting what type of calls they want to make, this can have a positive effect on their performance and job satisfaction (Sisselman & Whitt, 2009). One way of doing this by taking into account the preferences of the call agents (preference-based routing), which in the case of this project are the private bankers to whom the leads are sent. One can argue that including the teams as part of the data modelling already includes some implicit preferences of the bankers, in the sense that feedback is only received for calls that the bankers have accepted.

To some extent this might be true, but there might not always be a strong link between preference and performance. While a full study of the factors affecting the bankers' decision in accepting or rejecting a lead is a project of its own and out of scope of this research, the explicit preferences of the bankers for certain types of triggers and leads can be investigated on a small scale. Data about these preferences is not readily available, and as a result, it was decided to perform a small survey with participating bankers in order to collect such data. The methodology of the survey will be introduced in the following chapter 3 and the results will be discussed in chapter 5. Next, the relevance of the GDPR to this project and the measures taken by ING to ensure compliance are investigated.

2.2 On GDPR Compliant Data Processing

The increased use of AI in society as well as new regulations such as the GDPR have brought upon certain considerations when using data-driven algorithms. As a result of such considerations, there have been calls for the study of long-term effects of such 'datafication' (Newell & Marabelli, 2015) and for transparency and accountability for AI (Wachter, Mittelstadt & Floridi, 2017). As a result, any design of data models being made from now on must consider compliance with the data-related regulation, this applies to what kind of data is collected, how and for what purpose it is processed and what kind of decisions are made as a result. Therefore, it is important to consider how this regulation applies to automated lead management in particular, which is the goal of the data modeling in this project.

In order to answer the first subquestion then, an overview of the important points in the GDPR is provided, followed by its relevance to this project and the measures taken to remain compliant. First, some GDPR terminology relevant to the project needs to be defined:

1- Personal Data: the GDPR only applies to situations where processing of data includes personal data. The definition of personal data is very broad, and can be applied to any data by which a natural person (and thus not a corporation or other legal entity) can be uniquely identified either directly or indirectly by some reference. Examples of this could be a phone number, address, customer data etc. Customer data specifically applies to this project, therefore there is a definite need to consider the requirements of the GDPR.

2- Processing: refers to any operation(s) on personal data or sets of personal data, either manually or by automated means. Including but not limited to collection, storage, adaptation, alteration, destruction, etc. By this definition. the collection, aggregation and usage of data in the form of triggers which are subsequently used for data modelling in this project is then a form of processing.

3- Controller: natural or legal person or any authority which decides alone or in consultation with other relevant bodies on the purposes and means of processing personal data. In this case, this would be ING.

4- Processor: natural or legal person or any authority which performs processing of personal on behalf of and only by instruction of the controller. In this case it is the same as the controller, because it is an in-house analytics department. The distinction between the controller and processor becomes more relevant in cases where companies outsource analytics operations.

5- Profiling: any form of processing of personal data in order to estimate or evaluate certain attributes relating to a natural person, including but not limited to performance at work, economic situation, health, location and movement or reliability and behaviour. In the case of this project, in the creation of the triggers, personal data is processed and used in decision making in order to identify which leads must be prioritized (which customers must be contacted) each week. This can relate to many different topics depending on the nature of the triggers, as explained in section 1.1.

There are six legal grounds in the GDPR based on which data processing can be performed:

1- Consent: the data subject has given consent for use of their personal data for one or more purposes. In the case of data-driven marketing, it would be fairly difficult to use prior consent every time data is used, given the trial and error nature of data model development.

2- Contractual necessity: data processing is necessary for the performance of a contract in which the data subject is part of. For example, if somebody signs up to receive a certain newsletter, their email address (or in the case of physical newsletters, house address) must be processed for delivery.

3- Legal obligations: data processing is necessary for compliance with legal obligations to which the controller is subject to. For example, the bank may be required by law to provide some account information to the tax authority (Belastingdienst) in order to ensure tax obligations are correctly fulfilled.

4- Legitimate interests: data processing is necessary for pursuing the legitimate interests of the controller and/or a related third party, except for cases where such interests are overruled by the interest and/or fundamental rights and freedoms of the data subject which demand protection of personal data. This category is the most flexible in terms of interpretation, and any such ruling will be subject to the evaluation done by the controller. More on this further below.

5- Public interests: data processing is necessary in a case involving public interest or in a task in the official authority vested in the controller. An example could be data exchange between the bank and the police force for the purpose of aiding criminal investigations.

6- Vital interests: data processing is necessary for protecting the vital interests of the data subject or other natural persons. For example, in the case of the healthcare industry this could be processing someone's health related data, or in the particular case of terrorist attacks, data exchange between the banks and the police can also fall under this category.

In the case of financial institutions, most have opted for the legitimate interests option. In a direct quote from the Dutch Minister of Finance (Tweede Kamer, 2018): "In beginsel is op grond van de AVG geen aparte toestemming nodig van de klant voor het gebruik van klantgegevens voor direct marketing door de bank zelf ten behoeve van eigen, bestaande klanten"

And translated to English: "In principle, the GDPR does not require separate permission from the customer for the use of customer data for direct marketing by the bank itself for its own existing customers"

This quote was made in relation to the data used by an app from another bank, but in principle it is basis for the position that all financial institutes have opted for in the Netherlands. In the case of ING, personal data is categorized into four levels, in terms of increasing importance. Level 1 data includes basic data which are always required for normal services, like name or address. Level 2 data includes product and/or agreement data, such as the types of products a customer has. Level 3 data includes product use data, for example the aggregated input/output of an investment account over a period of time. Level 4 data includes detailed data such as specific transactions and/or counterparties unique to a customer. Only for level 4 data prior consent is required for processing. As a result, starting on May 25th 2018 (the date on which the GDPR went into effect in the EU), the use of all level 4 data was stopped for data-driven direct marketing. In the case of the REVA, this largely meant two changes:

1- Personal data for customer acquisition could no longer be used, which significantly reduced the number of prospective customer (by half)

2- Specific transactions and specific counterparties for transactions could no longer be used, which reduced the number of triggers. Note that, aggregating transactions over a time period without specific details (for example, an increase in balance of n euros over the last month) is not considered personal data, as it does not contain any sensitive information unique to the customer.

Next, key issues in terms of data agent rights according to the GDPR must be defined:

1- Right to be informed: in this sense there are two cases which are distinguished, if data is collected directly from the data agent, they should immediately be informed. If it is indirect, they should be informed within a reasonable time period, up to a maximum of one month. In the case of this project, in either case the data agent by their accordance with the privacy statement which they are presented and agree to. This statement is also being updated to make the means of data processing fully transparent.

2- Right to be forgotten: this clause arises a landmark case of Google Spain SL, Google Inc v Agencia Española de Protección de Datos, Mario Costeja González. In the aforementioned case, the data agent requested a correction of their out of date data which linked a Google search of their name to a then re-paid public debt. This led to a bad reputation for the data agent and made them have trouble finding employment. As a result, it is now required that any data controller must erase any personal data at the request of the data agent, unless there is any legal ground for keeping the data and/or processing it for the controller.

3- Right to object: based on this clause, the data agent at any time reserves the right to object to processing of their personal data, and the controller must stop the processing at the request of the data agent, unless the controller has compelling legitimate grounds upon which their interest outweighs the interest and rights of the data agent.

4- Right to explanation: this is a much contested part of the GDPR. First and foremost, it is important to note there is no such thing as a right to explanation in the text of the GDPR. However, in recital 71 which is about specific condition with regards to profiling, it is mentioned: " *In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision.*"

The purpose of any such explanation seems to be transparency. Transparency itself can be defined in different ways (Theodorou, Wortham & Bryson, 2017):

1- Lack of deception: that is not pretending a model is not a robot when in fact it is.

2- Reliability: that is not withholding information with regards to the limitations of any designed model.

3- Unexpected behaviour mechanism: that is the model being able to identify its own unexpected results.

4- Decision making exposure mechanism: that is being able to explain why a decision was made and how would it not have been made. The focus of the recital is on this definition, as indeed the idea is to be able to explain the workings of the model for how a decision is arrived at.

In many cases, explainability and model performance are competing values. In Figure 6 below, some of the most popular machine learning algorithms today are ranked in a matrix on general performance and explainability:

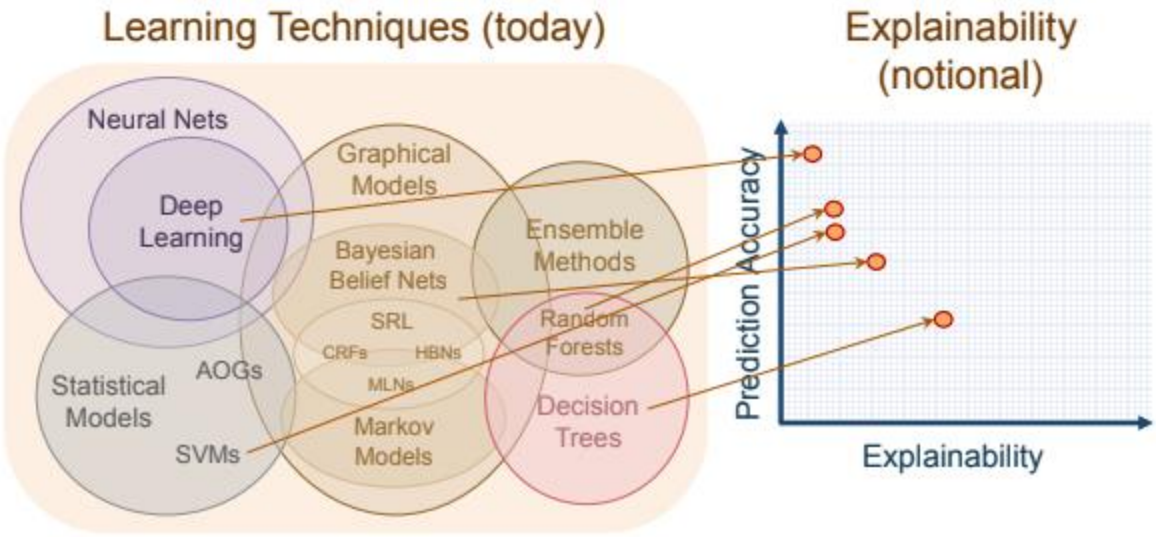


Figure 6: Prediction Accuracy vs. Explainability. Reprinted from “Explainable Artificial Intelligence (XAI)” by David Gunning - Defense Advanced Research Projects Agency (DARPA), 2017

In the above report, several modified approaches are also presented to the more popular high performing algorithms, which mostly have some form of modification for better explainability. A technical explanation of the mentioned approaches is beyond the scope of this report, but interested readers are encouraged to review the original report by DARPA.

There are some who argue against demanding full explanation of every algorithm used. Too much focus on an explanation framework can lead to an enforced and impractical transparency framework, parallel to the existing impractical data consent mechanism. Therefore, a better remedy is to take steps to simply make better machine learning algorithms that are less prone to discrimination and bias based on their input data, rather than come up with ways to explain the exact inner workings of algorithms to data subjects (Edwards & Veal, 2017). The algorithm here does affect which customers are contacted at what time and in a lot of cases the results of these contacts lead to the sale of financial products such as loans or mortgages, which in a broader sense are examples of allocation of capital, as explained earlier. These decisions have long-term consequences for society as a whole, and it is important the decision making process is fair and unbiased and leads to the best collective outcome.

In the end, efforts can be taken to make decisions more interpretable to both the users of the models and the data agents. The specific methodology, as explained earlier in the DARPA paper, depends on the algorithm used. In this project, based on the final model selected, measures will be implemented to make the decisions more interpretable. For details, please refer to section 5.2.

In a lot of clauses the GDPR mentioned above, the “unless the interests of the processor overrules that of the data agent” clause appears. It is natural then to continue with how this comparison can be made. To complement the GDPR regulation, the recommended procedure for evaluating such questions is a Data Protection Impact Assessment (DPIA). The DPIA is not concretely defined in the GDPR, but there is a guidelines document provided by the Data Protection Working Party, which is an independent advisory to the EU. According to the guidelines document: "*is a process designed to describe the processing, assess its necessity and proportionality and help manage the risks to the rights and freedoms of natural persons resulting from the processing of personal data4 by assessing them and determining the measures to address them. DPIAs are important tools for accountability, as they help controllers not only to comply with the requirements of the GDPR, but also to demonstrate that appropriate measures have been taken to ensure compliance with the Regulation. In other words,* ***a DPIA is a process for building and demonstrating compliance***."

In the case of this project, the activities fall under customer intelligence (CI) at ING. There procedure followed is based on risk analysis and is called a balancing test. The balancing test is a procedure which aims to compare the interest of ING against that of the data agent with the goal of preventing disproportionate negative impact on anyone as a result of personal data processing. The test is based on five criteria:

1- Remoteness: refers to the distance between the initial purpose of data collection and any potential further processing.

2- Context: refers to circumstances of data collection and the reasonable expectation of the data agents as to their further use.

3- Nature of the data: refers to the exact data type.

4- Possible consequences: refers to potential impacts of processing, including but not limited to potential factual and emotional impact on the data agent.

5- Safeguards: refers to measures taken by ING to ensure fair processing and prevent any undue impact on the data agent.

Based on the results of the test per CI activity, the legal ground is chosen either as legitimate interest (green outcome) or potentially consent (yellow or red outcome). As mentioned earlier in chapter 1, governance of algorithms also needs to be considered. Loss of accountability and human oversight have been identified as important risks when using algorithmic decision making which complicate the governance of the process (Bejtullahu-Michalopoulos & Florin, 2018). Therefore, as an extension to the five criteria that are presented in the balancing test, interpretability can be added as well. Then, the 6th factor can be:

6- Decision Interpretability: refers to the extent by which a models decisions can be understood, interpreted and connected back to the key factors which led to the outcome

The addition of this factor would mean taking measures or at least recording to what level can the decisions made by the model be understood. This can help both the users of the model and in certain cases the data agents as well. Especially in cases where a decision based on the model does lead to a negative outcome for a data agent, it is of utmost importance that the decision making process can at least to some level be understood by the users of the model, in order for the required modifications to be made to prevent further future damage. Overall, the GDPR has made data controllers a lot more aware about personal data processing and its potential consequences. The summary of actions taken to remain compliant can be found below:

1- The usage of level 4 personal data is stopped for CI activities. Instead of looking at individual data points unique to a customer, non-specific aggregated data over a period of time is used. Additionally, all data used is relevant to the business provided by the company, and any kind of special personal data such as gender, sexual orientation, political and/or religious beliefs/etc are not used to reduce the probability of bias based on these factors.

2- Customers are given a clear pathway to opt out of all personalized marketing, respecting the customers right to object.

3- The privacy statement is updated to ensure full transparency with regard to personal data processing, respecting the customers right to be informed.

4- For each CI activity a balancing test is performed and accounted for resulting in a legal processing basis of either legitimate interest or consent as defined in the GDPR, respecting the DPIA requirements.

5- Depending on the algorithm chosen, measures will be taken to ensure some level of explainability and interpretability of the decisions made (see section 5.3.3 where this will be further discussed based on the results). This is done in line with the newly proposed decision interpretability factor of the balancing test with the purpose of helping model users and facilitating the governance of the algorithmic decision making process.

In the following chapter, the methodology of the research is presented and explained.

3 Methodology

Following from the theory in the previous chapter, in this chapter the technical methodology is presented, which make the case for the usage of predictive analytics. The theory of each of the three algorithms going to be used is explained. Furthermore, the metrics by which the performance of the algorithms will be assessed are presented, as well as the theory behind the hyperparameter tuning which will be used to select the best setting of each algorithm. In the final section, the setting of the survey which is going to be performed for extracting the preferences of the private bankers is explained.

The methodology applied to answer the 3nd sub-question will be predictive analytics. To be more specific, a machine learning algorithm will be used to learn from historical data of previous leads in order to be able to predict the potential outcome of the newly generated leads. The algorithm learns from so called 'features' of the data. Features are any possible data about the outcome to be predicted. In the case of this project, features are mostly going to be indications which are generated from data about a customer, the types of which is explained in section 1.1. The outcomes to be predicted are called labels. In this case, the labels are generated by aggregating the feedback from the private into either positive or negative (1 or 0). Once the minimum set of features is constructed from the data, exploratory modelling will be performed by testing multiple algorithms with minimum parameter optimization. The best performing algorithms will then be further optimized and analyzed and eventually the best performing configuration of the best performing algorithm will go into production for live testing.

According to a survey by popular data science platform Kaggle, Python and R are the two most commonly used tools for data science in industry, respectively (Kaggle, 2017). Both are also extensively used in the TPM faculty and the courses of the EPA program, as well as having large active communities online for support and a variety of free data science related packages which make many previously difficult tasks more intuitive. For this project, based on personal experience and consultation with the bank, the tool to perform analytics will be Python. For the design and testing of the algorithms, the sk-learn library was chosen as it is relatively straightforward to perform most of the required steps, has a variety of useful in-built functions and is well documented (Pedregosa et al, 2011).

3.1 Algorithms

In this section the theory behind the algorithms that will be used in the initial testing of the classification model is described. First, an intuitive explanation is provided, followed by mathematical equations. The problem at hand is treated as one of binary classification. That is to say, the goal is to predict whether a given a lead at a given time is going to be successful or not successful (1 or 0, these are the labels, as explained in the previous section). There are two common types of algorithms used for the task of classification in machine learning, which are discriminative and generative algorithms. The difference between the two is in the approach to the probability calculation.

Discriminative algorithms calculate the conditional probability of an outcome (Y) given certain features (X), that is: *P(Y|X)*. In essence, these algorithms model the decision boundary between different outcomes in classification. Generative models on the other hand, calculate the joint probability of the outcomes and features, that is: *P(X, Y)*. Subsequently, Bayes rule is used to make predictions for the conditional probability. In essence, these algorithms model the actual distribution of the outcomes in classification. Generative algorithms are also capable of generating likely X,Y pairs as well as making predictions. The choice between the two depends on many factors, but in general, it has been argued that when data availability increases, discriminative models tend to match or even outperform generative models in classification tasks (Y.Ng & Jordan, 2002).

Additionally, in this project generating likely pairs does not add extra value, as the features of the data are dynamically generated and assigned based on real-life events from the customers and may vary considerably over time. Considering all this and the fact that the training data size will keep increasing going forward, the three algorithms which are going to be tested for the project are all from the discriminative type.

There are arguments against trying different algorithms for the same problem. This argument is based on No Free Lunch theorems particular to machine learning which was introduced in the paper by Wolter (1996). There, it was argued that there is nothing to be gained by different learning algorithms on the same problem without specific assumptions about the data or problem. Nevertheless, while at this point of the project there were no assumptions to be made on the data, it was chosen to try 3 different algorithms based on generally perceived prediction accuracy (as also seen in Figure 6, noting that the exact relations on that graph may not be applicable to this problem), relative notional explainability and computational efficiency, to have at least done the analytical due diligence to try different approaches with the potential of obtaining significantly varying performance. Below, brief theoretical explanations per algorithm (in the context of classification) are given and where appropriate, referrals to more detailed sources are provided.

3.1.1 Logistic Regression

Logistic regression is a regression technique which uses the logistic function. The central mathematical theorem behind the logistic function is the logit, which is the natural logarithm of an odds ratio (Peng et al, 2002). An odds ratio of the events y and x, is defined as the ratio of the probability of event y in the presence and absence of event x. In mathematical terms, that is:

$$3.1.1 \quad logit(odds) = ln(\frac{p(y|x)}{1-p(y|x)}) = \beta_0 + \beta_1 x$$

$$3.1.2 \quad p(y|x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Where equation 3.1.1 shows the logit of the odds ratio between events y and x, which is modelled as a univariate regression. In such an equation, the first term is called the intercept or bias, and subsequent terms are pairs of inputs and their corresponding parameters. Equation 3.1.2 shows the probability calculation which takes the form the standard logistic function and is arrived at by taking the antilog of both sides of equation 3.1.1 and solving for the probability.

Note that a univariate regression is used for simplicity as an example, naturally with more features present, more input-parameter pairs will have to be added. The logistic regression models the probability of the default class using equation 3.1.2, and based on the probability threshold (which is set at 0.5 by default), it converts the probabilities to binary decisions in order to make predictions on which class each input belongs to. The parameters (β_0 and β_1) are to be estimated from the training data. There are multiple possibilities with regards to how these parameters are updated based on each training stance, usually making use of the prediction error in some form. A relatively simple example for any given parameter could be:

$$3.1.3 \quad \beta_n = \beta_n + \alpha(y - y_{pred})y_{pred}(1 - y_{pred})x_n$$

Where α is the learning rate, y is the label, y_{pred} is the predicted label and x_n is the corresponding input value to the parameter β_n. Note that is such cases, it is common practice to use an input value of 1 for the bias β_0 (Russel & Norvig, 2009). The exact settings of the model which is used for this project will be discussed further in the following chapter.

In terms of explainability, this algorithm is fairly white-box, in the sense that it can almost always be traced back and analysed exactly which factors led to a certain outcome and how a different outcome might have been arrived at. Therefore, it is reasonable to say it has high explainability. Due to the relatively simple and linear nature of the calculations, the computational efficiency of the logistic regression algorithm is relatively high compared to the other two.

For a more detailed account of logistic regression in the context of classification, interested readers are referred to the presentation on the topic by Stine (n.d.) from the Department of Statistics of Wharton Business school.

3.1.2 Random Forests

In order to explain random forests, first an explanation of decision trees is required. A decision tree is an algorithm that can perform both regression and classification tasks by learning decision rules. An example of a decision tree can be seen in Figure 7 below:

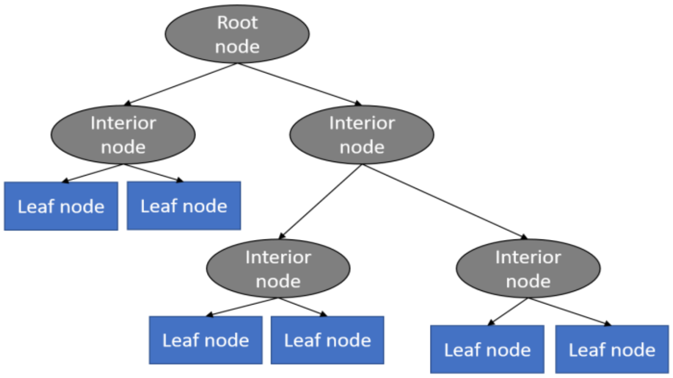


Figure 7: Representation of a typical decision tree

The key point in the case of decision is the order of placement of features along the tree, with the most important being on top (root of the tree) and from then onwards features with decreasing importance down to the leaf nodes. There are many ways to select this importance, with the two most common being by calculating information gain or the Gini index. The information gain is defined as the change in entropy of an outcome by including a certain feature. The concept entropy derives from information theory, but here a brief mathematical introduction is provided. For a set of possible values of an input and their corresponding outputs, the entropy is defined as:

$$3.1.4 \quad E(X) = -\sum_{x \epsilon X} p(x) \log p(x)$$

where p(x) is the probability of the outcome belonging to the default class given a condition on the input x. Using equation 3.1.4, first the entropy of the default label is calculated, and then the of the default label given every feature. Thus, the information gain of any feature F and default label T can be defined as:

$$3.1.5 \quad IG(F) = E(T) - E(T|F)$$

following from these calculations, the variables with the highest information gain can be put at the root of the tree. The Gini index is very similar to the information gain, but it doesn't contain a logarithmic calculation and is thus more computationally efficient. It can be defined as:

$$3.1.6 \quad GINI(X) = 1 - \sum_{x \epsilon X} p(x)^2$$

Furthermore, it has been argued that in the majority of cases (>98%), both methods yield similar results (Raileanu & Stoffel, 2004), thus making Gini the more common choice. In terms of explainability, decision trees are easily traceable and can be visualized to see exactly what condition caused a decision to be made at each step. One of the main weaknesses of decision trees is that they are prone to overfitting. When a model fits training data so well that it accommodates the noise/anomalies in the training as valid data points, it will have a detrimental effect on the models performance when presented with new data. This concept is called overfitting.

In order to mitigate overfitting among weaknesses of simple learning algorithms, the concept of ensemble learners needs to be introduced. Ensemble learners use collection of 'weak' learners like decision trees and use the aggregate result of all the underlying learners in order to act as a more robust learning algorithm. The random forest algorithm is in essence an ensemble learner which uses decision trees as its underlying weak learners for this purpose. There are more detailed factors to take into account when using a random forest, which will be explained further in chapters 4 and 5.

In terms of explainability, while in theory random forests are a collection of decision trees and should be highly explainable, in practice it is often not the case. This is due to the fact the trees used will usually get too crowded (or deep) to be able to trace or visualize efficiently and moreover there will be too many trees to feasibly look at one by one and trace back the aggregate result back to one leaf in one tree. This will be demonstrated in the following sections as the algorithm is implemented. Therefore, it is reasonable to say this algorithm has practically medium/low explainability. Due to the very large feature matrix upon which the calculations of each decision tree will be based on, the computational efficiency of this algorithm can be anywhere between medium to low, largely dependent on the number and size of trees in the ensemble.

For a more detailed account of random forests in general and in the context of classification, interested readers are referred to the original paper written by Breiman (2001), which introduced the ensemble random forest approach.

3.1.3 Multi-Layer Perceptron

First, an introduction of Artificial Neural Networks (ANNs) is required. ANN is a multi-layer construct consisting of an input layer, a minimum of one 'hidden' layer and an output layer. Within each layer, there are neurons which carry signals (after processing them) to the neurons in the following layer. A visual representation of such a structure can be seen in Figure 8 below:

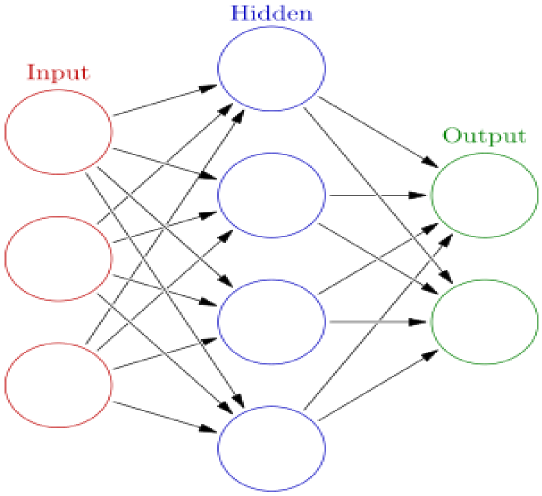


Figure 8: A representation of a one (hidden) layer MLP

A Multi-Layer Perceptron (MLP) is a type of feedforward neural network which can be used for both classification and regression. Figure 8 is representative of a feedforward network in the sense that the direction of signal propagation is always forward (there are no cycles or feedback loops). MLPs are fully connected networks, in the sense that each neuron in a layer is connected by a forward connection to every neuron in the following layer. Each layer (except from the input layer) has an activation function which does some sort of (linear or nonlinear) mapping of the input in the neurons. Examples of common activation functions are:

$$3.1.7 \quad f(x) = \frac{1}{1+e^{-x}}$$

$$3.1.8 \quad f(x) = tanh(x)$$

$$3.1.9 \quad f(x) = max(0, x)$$

Where the equations 3.1.7, 3.1.8 and 3.1.9 represent the standard logistic function, the hyperbolic tangent function and the Rectified Linear Unit (ReLU) function, respectively. The choice of function depends on the task and can also be done by trial and error. The most common choice is the ReLU function as it can handle most tasks efficiently, including sparse inputs.

The learning happens by updating weight parameters in each neuron based on the results of the output layer and their deviation from the actual outputs. There are different approaches by which the weights are updated, but most commonly some form of gradient descent is utilized. The idea of gradient descent is to use the gradient of a function (the cost function, see below) with regards to every weight in order to find the minimum value of the function (lowest error). The result of the gradient updates is then fed back through the network in order to update the weights at each previous layer, this process is called backpropagation. Going into the derivation of weight changes through backpropagation is beyond the scope of this report, but interested readers are referred to the paper by Sathyanarayana (2014). The goal is constant updating of the weights is to minimize a form of cost function. The cost function in the particular MLP implementation in this project is the log loss function (simplified for binary classification):

$$3.1.10 \quad l(x, y) = -(y \log p(x) + (1 - y)(1 - p(x)))$$

Where y is a binary label (0 or 1) indicating whether an input x is correctly predicted, x is an input and p(x) is the probability calculation of the model that input x belongs to a certain label. Again, there are many other factors to consider in the implementation of this algorithm, which will be explained in the following chapters. In terms of explainability, given the depth and complexity of the calculations in this algorithm, it is fairly difficult to trace back single decisions to their underlying factors, and it is reasonable to say the algorithm has low explainability. The computational efficiency of neural networks in general is rather low, while of course being dependent on the number of features, hidden layers and the number of nodes in each layer (and to a lesser extent, the activation function in each layer). In Table 2 below, an overview of the three algorithms in terms of expected performance, explainability and computational efficiency is provided:

Table 2: Overview of the three Algorithms

Algorithm	**Expected Performance**	**Relative Explainability**	**Computational Efficiency**
Logistic Regression	Medium	High	High
Random Forests	Medium/High	Low/Medium	Low/Medium
Multi-Layer Perceptron	High	Low	Low

3.2 Algorithm Metrics

In this section, the metrics used for the evaluation of the different algorithms will be defined, as well the cross-validation technique used to ensure more robust performance.

3.2.1 Precision, Recall, F1 Score and Accuracy

The metrics in this section can be calculated per label and on average as a form of evaluation for the performance of the model. Precision is defined as the ratio of correctly labeled items over the total items assigned to a label by the model, whereas recall is defined as the ratio of correctly labeled items over the total number of items selected by the model which had that label. Note that, both precision and recall will have values in the range between zero and 1 (100% correct predictions). Figure 9 below provides a clear explanation of the difference between the two metrics:

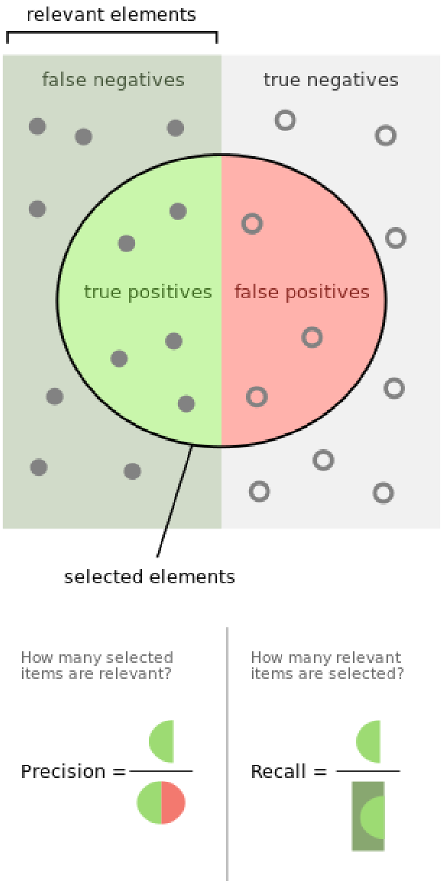


Figure 9: Visual representation of precision and recall

The F-measure of f1-score is defined as the harmonic mean of precision and recall:

$$3.2.1 \quad F1 = (\frac{precision^{-1} + recall^{-1}}{2})^{-1} = 2.\frac{precision.recall}{precision + recall}$$

Furthermore, accuracy can be defined as:

$$3.2.2 \quad Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

Where tp, tn, fp and fn represent true positives, true negatives, false positives (Type I error) and false negatives (Type II error), respectively. Which metric is most important is an aspect to the problem itself (Hand & Christen, 2018). This will be discussed further when the models are evaluated.

3.2.2 k-Fold Cross-Validation

Cross-validation is a model-validation technique used to gain an understanding of how the performance of a model would generalize on an independent data-set (Kohavi, 1995). This is done to detect potential overfitting or inconsistencies in model performance. The general idea behind cross-validation is partitioning data into complementary subsets, using certain subsets as training and others as validation data and aggregating performance metrics across different subsets, in order to achieve a more robust estimate of model predictive performance (Seni & Elder, 2010). There are two main types of cross-validation, exhaustive and non-exhaustive. Exhaustive cross-validation techniques try all possible combinations of splits for the data and as a result, as data availability increases, lead to an unfeasible amount of calculations. Non-exhaustive methods test only certain combinations of data splits based on predetermined factors.

K-fold cross validation is a type of non-exhaustive method, which randomly divides the data into k equal parts, fits the model by using one of the k partitions as a validation data-set and the rest as training, and iterates through the partitions until each partition has been used for validation exactly once. So for k folds, there will be a total of k fits. The output result will then be the performance of the model (with a chosen metric) aggregated over the k fits.

Having defined the performance metrics and the method used to estimate their robustness, the following section provides the theory another important step in developing a predictive model, which is the fine-tuning of the model hyperparameters.

3.3 Hyperparameter Tuning

Hyperparameters are parameters of the algorithm which have to be set before fitting the data, whereas most other model parameters are learned from data during training. Hyperparameters can be numerical or categorical, dependent on other hyperparameters and can be a determinant factor of both the performance and runtime of the model (Claesen & De Moor, 2015). The number and type of hyperparameters is dependent on the algorithm, for a full list of parameters which are considered per algorithm for this project please refer to Table A-5 in Appendix I.

Hyperparameter tuning is the process of choosing the optimal hyperparameters for the algorithm which is being used. There are many analytic approaches to hyperparameter tuning, in the following section the ones that will be implemented in the case of this project are introduced.

3.3.1 Grid Search & Random Search

The standard way to perform hyperparameter tuning is Grid Search. In this method, first a range of feasible values (usually it is reasonably easy to define this range) must be selected per hyperparameter. The combination of hyperparameters across these values is called the grid. For an example, an algorithm with n hyperparameters having m possible values each has an n times m grid (note that often times each hyperparameter has a different number of possible values, the assumption of the example is for the sake of simplicity). The Grid Search is defined as an exhaustive search through all possible hyperparameter combinations. Typically, this is accompanied by computing cross-validation scores of the model per setting, and the hyperparameters of the model setting with the highest cross-validation score are selected as optimal (Hsu, Chang & Lin, 2003). Note that, as it is an exhaustive search algorithm, the computational costs can become significant when the number of hyperparameters and/or their range of values increases.

A slightly more refined method is the Random Search. Similar to before, first a grid of hyperparameter values is constructed. The random search will randomly test a defined x number of possible candidates from the total possible n times m combination (keeping the previous example in mind). Due to the random selection, the idea is that it generally covers most of grid, yielding similar performance to the grid search, while being much more computationally efficient. Of course, depending on the number of samples randomly drawn, there is a risk that the whole grid is not covered (or the global optimum is not reached). To have a measure of this risk, a probability calculation can be performed (Zheng, 2015). If we assume that at least 3% of the configurations (taking 1% as too pessimistic and 5% too optimistic) in the grid are within the vicinity of the global optimum, then:

$$3.3.1 \quad p(miss, n) = (1 - 0.03)^n$$

$$3.3.2 \quad p(hit, n) = 1 - p(miss, n) = 1 - (1 - 0.03)^n$$

$$3.3.3 \quad p(hit, n) > 0.95 \rightarrow 1 - (1 - 0.03)^n > 0.95 \rightarrow n > 98$$

Where p(miss,n) shows the probability of every one of the n randomly and independently drawn samples missing the vicinity of the global optimum and p(hit,n) shows the probability of at least one candidate out of the n drawn samples being representative of the global optimum. Equation 3.3.3 shows the solving of an inequality whereby with a 95% confidence it can be said that regardless of the size of the grid, if more than 98 samples are drawn, the optimum found will be representative of the global optimum.

Here, it is proposed to define a relatively extensive grid and use a reasonable number of configurations (as calculated above) to perform a random search. Subsequently, define a much smaller grid based on the optimal parameters from the random search, and perform a grid search on this much smaller grid for a more robust result than only using a random search and a more computationally efficient result than only using a grid search. The details of this procedure are explained in chapter 4. In the next section , the methodology to be employed for the banker preference survey is introduced.

3.4 Banker Preference Survey

In this section it is aimed to explain the method by which an insight is to be gained into the explicit preferences of the banking teams in each region for different kinds of triggers. To do this, a survey is sent out containing ten high performing triggers (found in Table 6 in section 5.4) of different kinds for the bankers to rank in terms of preference. Due to the high total number of teams, the survey will be sent to the managers of each region in order to do with all the teams inside of the region. Therefore, the result will be one ranked list per region which can further be analyzed.

In order to reliably extract the preferences, a pairwise selection method is used. That is:

$$3.4.1 \quad Pairwise(10) = \binom{10}{2} = \frac{10!}{2!(10-2)!} = 45$$

$$3.4.2 \quad Score(T) = \frac{Count(T)}{9}$$

Where the first equation shows the total number of pairwise questions required, given that repetition is not allowed and order is irrelevant. The second equation shows the way the preference ranking is inferred, which is by using a score calculated by dividing the number of times any trigger T was chosen over the total number of times any one trigger can be selected. In this way, each trigger will achieve a preference score between 0 (minimum) and 9 (maximum). The results of the survey are presented and discussed in chapter 5. Having discussed the methodology of data modelling, evaluation and the survey, in the following chapter 4 the initial steps for the data modelling and evaluation are explored. The data models are created, tuned, validated and compared. Based on the results of these steps, the best model is then selected for testing in chapter 5.

4 Exploratory Data Analysis & Modeling

In this chapter, first the training dataset will be examined, as well as explaining the method by which features will be extracted (for an explanation of any technical terms appearing in this chapter please refer to chapter 3). During the examination of the dataset, considerations about extracting un-aggregated personal data as features need to be considered. Following from that, the model generation process in using the sk-learn library is explained, with code snippets being provided where necessary. Additionally, the addition of certain other data as added features will be examined, with the main deciding factor being whether they help improve the predictive accuracy of the model.

After the selection of the features, the process of parameter tuning for all three algorithms is explained and the results of said tuning are used to select the best setting of each model. In the final step, a closed test is performed between the three algorithms on newly generated data from one week in order to have a comparative measure of performance on live data, and based on the results, the best algorithm is selected for implementation and testing in chapter 5.

4.1 Data Exploration

The current dataset selected (which is about 20000 rows) consists of a unique identifier per customer, the top 5 triggers that were assigned to the lead for that customer and outcome of the contact. There are 191 unique triggers in use right now. It is worth noting these features are not static, and over time more features will be added which are defined by staff and approved as being relevant. Therefore, it is an important consideration that the model does not run into errors when encountering new features when it goes into testing/production. To overcome this, a re-index function is used, which will be explained further below in section 4.2. As a starting point, the number of each label is investigated in order to check for label imbalances:

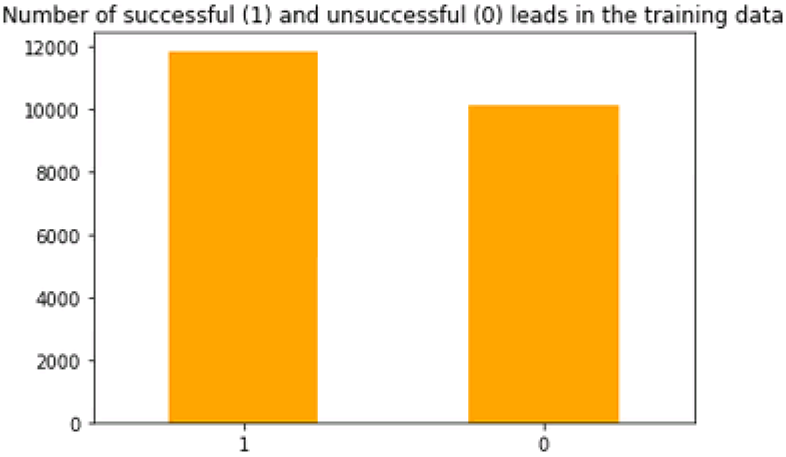


Figure 10: Frequency of each label in the training data

There seems to be a considerable number of each label available, which means there is not a re-balancing issue to worry about.

Looking at the features, the most common appearing triggers in all the leads and in the successful leads can be investigated:

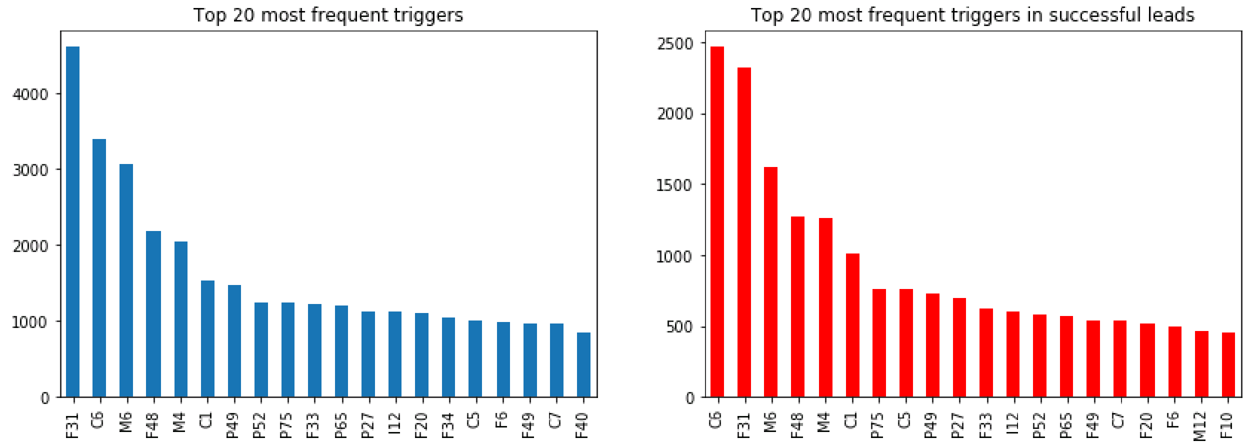


Figure 11: Frequency of triggers in certain leads

It can be seen that eighteen of the top 20 most frequent triggers are also in the most successful ones, which supports the idea that more frequent triggers are sent more often. This is not a surprise given the scoring method used to select the top five triggers, as explained in section 1.1. The triggers seem to be a balanced mix of indicators of financial capital availability, compliance, services and products. The fact that the services and products triggers appear in this result is encouraging, and supports the idea that the call center is really for customer service and not just generating revenues. The presence of the M6 and M4 triggers is a countermeasure against a self-fulfilling prophecy scenario where the same customers (or same type of customers) keep being called.

While appearance can be a measure of relevance for a trigger, a better measure for performance could be the conversion rate. Conversion rates can be misleading in the sense the triggers that appeared on one a few times in successful leads would have a 100% (or very high) conversion rate. In order to mitigate this, a subset of the forty most frequent triggers is investigated, with the twenty triggers with highest conversion rates being selected:

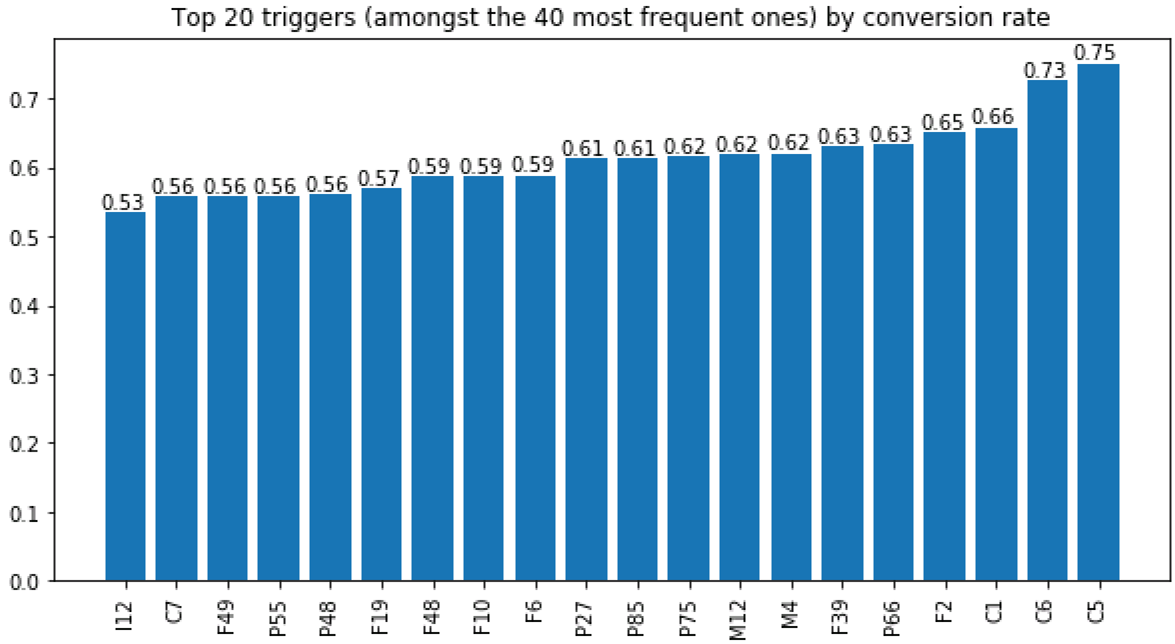


Figure 12: Frequent triggers with the highest conversion rates

The top three triggers which have considerably higher conversion than the others are all of the legal compliance type. Note that these are technically the same signal, just being reminded at different time steps. The number one trigger shows that this legal requirement is usually fulfilled at the eleven month mark (one month before the one year deadline). This leads to the idea that the bankers are well motivated in order to make sure the private banking service remains within the requirements of Dutch law, both to be a compliant bank benefitting the customers and of course to avoid potential repercussions such as fines by the regulatory institutions. Additionally, the indicators of customer not being contacted for a while are also there, as well as indicators of financial capital.

Thinking of additional measures, age and location come to mind. In terms of age, based on discussions with sales staff, they do not believe it to be influential on whether a sale is made or not. This can be investigated in the historical data:

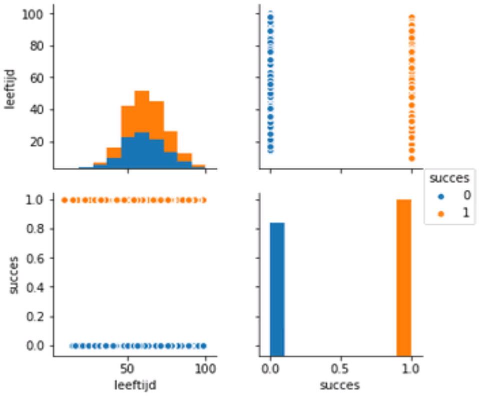


Figure 13: Pairplot between age and outcome of contact

Based on the relatively identical distribution of age over the two outcomes, it seems that indeed it is not influential on the outcome of the contact. It should be noted that this does not guarantee complete irrelevance of the age, as there may be interactions between age and other features present in the model. However, considering that there are already triggers which consider some of these potential interactions and the fact that extracting age as a lone feature constitutes using un-aggregated personal data, the addition of age as a numerical feature was deemed redundant. Additionally, it can be seen the all private banking customers seem to be in the above forty age range, which is an indication of mid/late career individuals. In terms of location, due to the business decision made to only send leads of a region to teams within that region, the addition of the team ids is technically a proxy for location as well, as each team only operates within a specific region. Therefore, insights can be gained with regards to certain types of leads selling better in certain regions (or not).

Excluding age and geographical location from further analysis, the triggers are left as features. These are all categorical values in five columns in the data. When dealing with most machine learning algorithms, categorical variables will have to be converted to numbers. There are two main approaches:

1- Categorization: This means simply assigning a unique number to each unique value of the categorical variable. This makes sense when the variable is ordinal, for example the days of the week. If Tuesday is assigned two and Monday as one, the ordinal relationship is natural and not induced. This however may be problematic when dealing with normal categorical variables which have no natural order, very much like the triggers at use in this project. The problem arises when the algorithm assumes order between the different values of the categorical variable and learns from patterns in the numbers that in a real-world context are meaningless. Then, the following technique is more suitable.

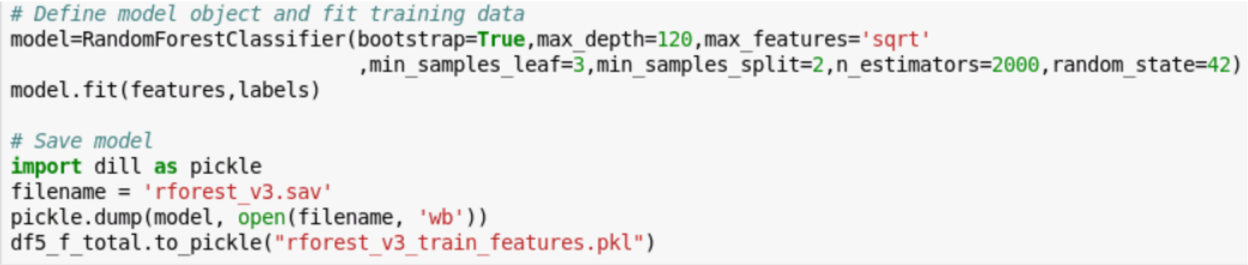
2- One-Hot Encoding: This technique entails treating each value of the categorical variable as a binary variable itself, that is to say transform a categorical variable with n unique values into n binary variables (1 or 0). Naturally, this leads to a relatively large number of features, and in this case due to the fact that a customer has a maximum of 5 triggers available, relatively sparse matrices.

The initial testing of the 3 algorithms is done using one-hot encoding on the unique values in the 5 columns corresponding to the top five triggers. It is worth noting that these columns themselves have some form of order in their values. That is based on the formula explained in section 1.1. That is to say the same trigger appearing on the second position out of five for a customer, has had a different historical conversion than one appearing on a fifth position out of five for another customer at another time. The choice is then to treat these differences, which are caused by the previous scoring system, as different features or ignore this difference and evaluate simply for the presence of a trigger in a lead, regardless of the position.

The argument for considering the difference would be to avoid potential information loss and the argument against it would be that it makes the feature matrices unreasonably sparse (which leads to unreasonably long computation times) and that the case in which this phenomenon occurs are too little to make a big difference in the performance of the model. To make a decision, we fit the three models in both cases (with default hyperparameters), and compare the results and runtime. The results are compared on the basis of average accuracy score over a 5 fold cross-validation per algorithm. In the following section, the model generation and the results per setting are discussed.

4.2 Model Generation & Results

As mentioned earlier, the making of the models will be done using the sk-learn library in Python. Each of the algorithms will be defined as objects, with the (hyper)parameters being defined as inputs. After the feature extraction from the data, then the feature matrix is fitted to the model object. The fitted model object can then be saved and later loaded again to be used for predictions with features extracted from new data. Example code for these procedures are provided in Figure 14 below:

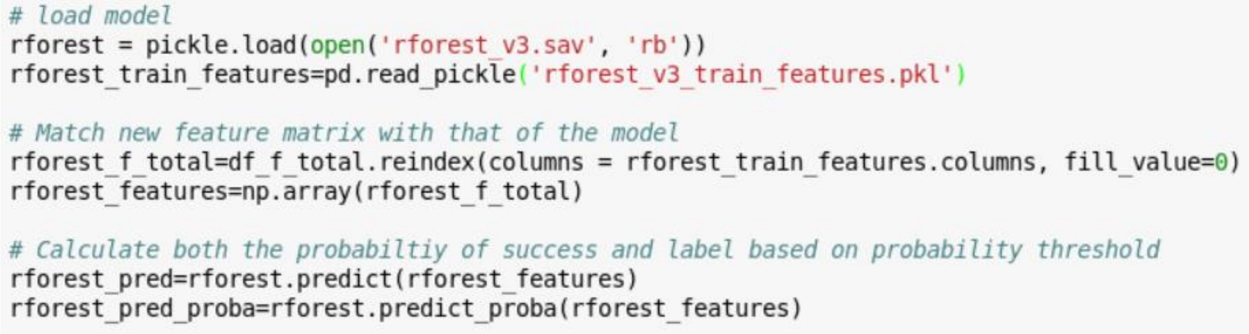


```
# Define model object and fit training data
model=RandomForestClassifier(bootstrap=True,max_depth=120,max_features='sqrt'
                              ,min_samples_leaf=3,min_samples_split=2,n_estimators=2000,random_state=42)
model.fit(features,labels)

# Save model
import dill as pickle
filename = 'rforest_v3.sav'
pickle.dump(model, open(filename, 'wb'))
df5_f_total.to_pickle("rforest_v3_train_features.pkl")
```

Figure 14: Example code to create, train and save model object

The model is defined as a random forest object, with the parameters selected. Then the features and labels which are extracted from the data are fit to the model. The fitted model is then saved for future use. The reason the model feature matrix is saved as well is due to the way new features are dealt with. When dealing with live data, there are two possibilities with regards to new features. Either the new data does not have the features that the model has, or the new data has new features that the model has not seen before. In the first case, the feature columns are simply added with values of 0 (indicating absence of that feature). In the second case, the column is dropped in order for the new feature matrix to match the one the model was trained on in terms of number of columns. When the model is re-trained with the labelled data, these features will be learned. This operation can be done with a column match function. An example code for loading a previously saved model to predict with new data features can be found below:



```
# load model
rforest = pickle.load(open('rforest_v3.sav', 'rb'))
rforest_train_features=pd.read_pickle('rforest_v3_train_features.pkl')

# Match new feature matrix with that of the model
rforest_f_total=df_f_total.reindex(columns = rforest_train_features.columns, fill_value=0)
rforest_features=np.array(rforest_f_total)

# Calculate both the probabiltiy of success and label based on probability threshold
rforest_pred=rforest.predict(rforest_features)
rforest_pred_proba=rforest.predict_proba(rforest_features)
```

Figure 15: Example code to load model, match feature matrix and make new predictions

The re-index function used in the above code can both add the features that the new feature matrix is missing (giving them 0, as defined in the fill_value input) and remove features that are not known to the model (by comparing column indices of the new feature matrix with that of the model). Note that, the creation of the specific feature matrices are omitted intentionally, as they involve confidential data and in any case are project dependent, so their omittance does not take away from the point of this section.

As discussed in the previous section, the model can be made with either a much bigger feature matrix involving the same triggers in different positions as separate features, or a much more compact one by only considering the presence or absence of a trigger in a lead. Treating the different position as different features yields a feature matrix of 778 columns. The performance results and runtime of the three algorithms in this setting can be seen in Figure 16 below:

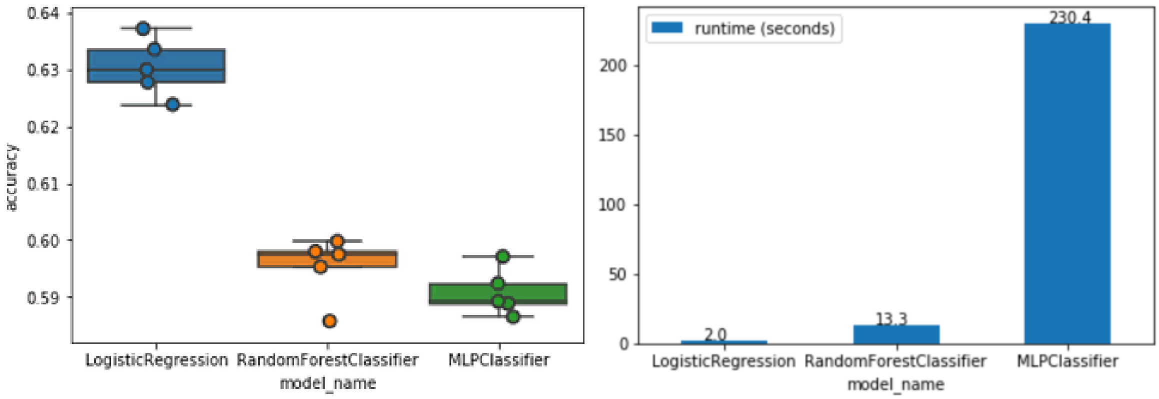


Figure 16: Runtime and performance of the three algorithms with the initial feature matrix

The logistic regression is the best performing and fastest algorithm, being more than one hundred times faster compared to the perceptron. The exact results of the evaluation can be found in Table A-1 in Appendix I. To ignore the position of the triggers, the previous feature matrix is joined horizontally, with the same triggers of different positions being aggregated by using a maximum function. This essentially evaluates the presence of the trigger in any of the five positions by calculating the maximum element out of a binary array of length five containing the presence of the element in each column. So the resulting column corresponding to the trigger will have a value of one if the trigger is present in any position in the lead, and otherwise it will have a value of zero.

Using this method, a feature matrix of 175 columns is constructed. The results and runtimes of the three algorithms with this feature matrix can be seen in Figure 17 below:

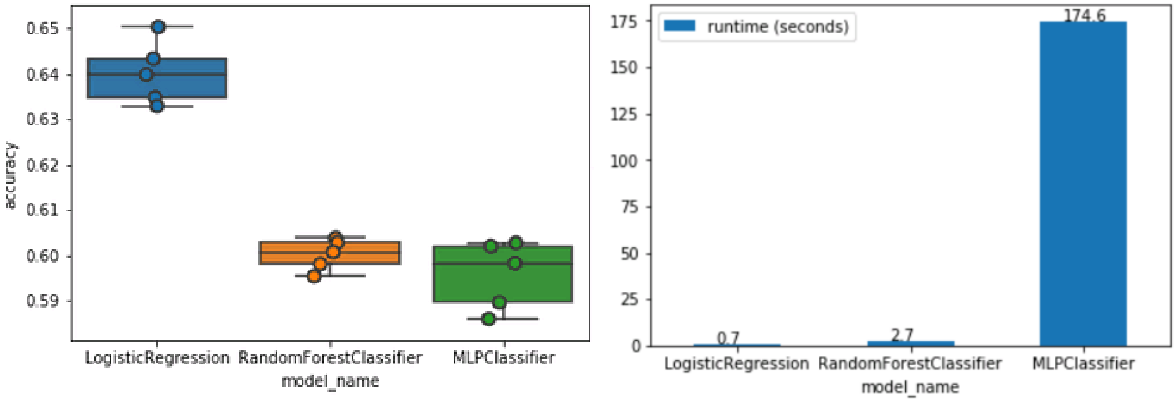


Figure 17: Runtime and performance of the three algorithms with the compact feature matrix

Again, the logistic regression is the best performing algorithm and all three algorithms are considerably faster as a result of the more compact feature matrix. The exact results of the evaluation can be found in Table A-2 in Appendix I. Comparing the two settings, it seems that the compact matrix yields slightly better performance with more efficient computation times, therefore there doesn't seem to be much information loss to worry about in this case. Having the initial features selected, next possibilities of using additional information are investigated.

4.2.1 Addition of Team Data

As there are more than one team in each region, it is worth investigating the effect of each team on the success of the leads. While the survey will be performed to gain an insight into the preferences of the teams with regards to the types of customers and triggers which they receive, there is already a possibility to investigate the performance of the teams with regards to the different types of leads. The team id will be used as the identifier of each team to which the leads have been historically sent to. This id can have anywhere between two to eight digits depending on the team in question. Investigating this id, it was found that there are certain values which are duplicates representing the same team, with extra zeros in front (example: 0010 & 10, two different ids but actually representing the same team). In order to overcome this, it was found that selecting the five right-most digits per team instead of full values yields a unique identifier per team. Therefore, this will be used as a feature representing the teams. This feature will also be treated like the triggers, with one-hot encoding performed to gain a set of columns which can be added to the feature matrix. After this addition, a matrix with 380 columns is obtained.

To evaluate to see if this can have an effect on the outcome, two figures are looked at. The aggregate historical conversion of teams in all regions and the aggregate performance of teams in the five most active regions (ones that have received the most number of leads in the past year) over the twenty most frequent triggers, as seen in Figures 18 and 19 below, respectively:

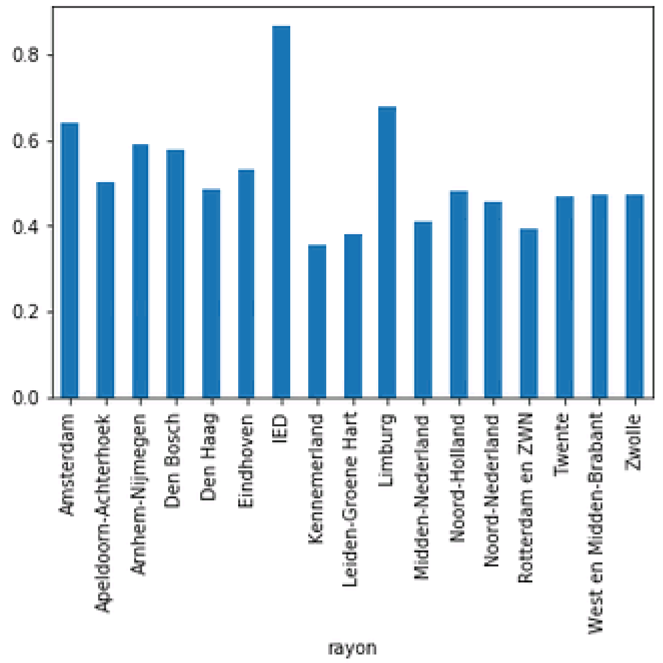


Figure 18: Aggregate team performance per region

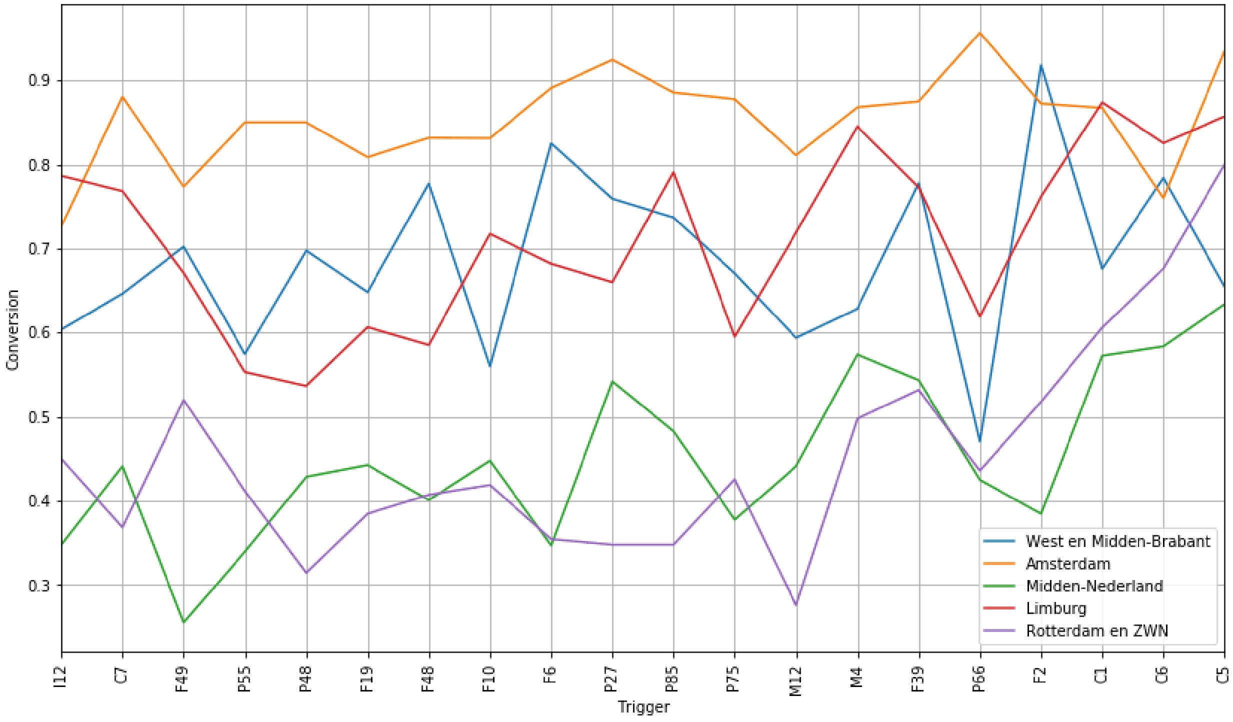


Figure 19: Aggregate performance of five most active regions across twenty highest performing triggers

Figure 18 shows the difference in overall performance between different regions. Figure 19 shows that while in some regions (like Amsterdam) performance is relatively stable across a variety of triggers, others (like Den Haag) seem to have a considerable variance in performance depending on the types of triggers in their leads. These insights suggest that adding the team ids can indeed help the predictions. The performance results and runtimes after the addition of the team ids can be found in Figure 20 below:

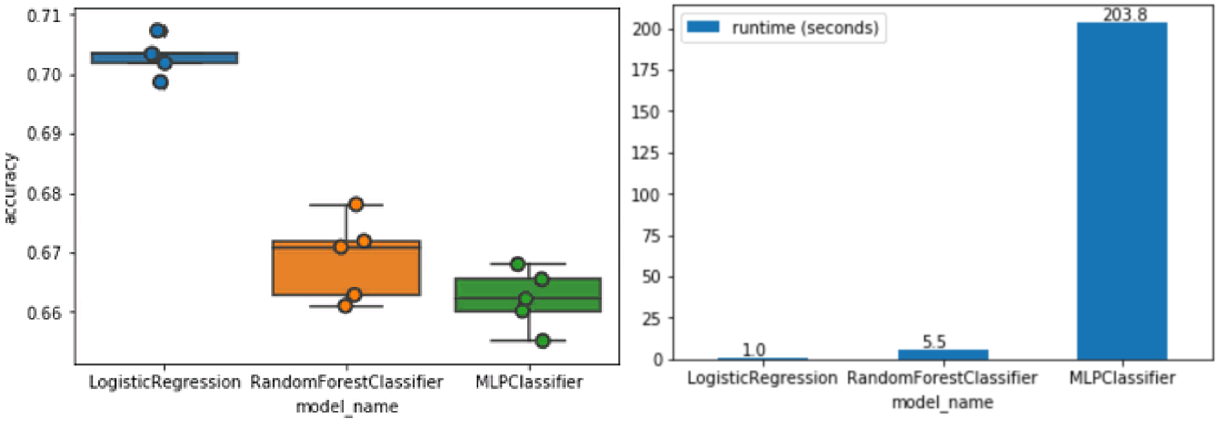


Figure 20: Runtime and performance of the three algorithms after addition of team ids

Looking at the performance of the models, the addition of this data boosts the performance of all three algorithms by about 7%, with the logistic regression remaining the best performing one. The runtimes are slightly longer, which is due to the expansion of features matrices by the team ids which are encoded, adding one column per team. The exact results of the evaluation can be found in Table A-3 in Appendix I.

4.2.2 Addition of Customer Potential

Another factor that can be used as a proxy of how bankers perceive customers is a feedback metric which is filled by the bankers after calling a customer. This metric can take a number of categorical values. In order to use this metric as features, first each categorical value was mapped to either a low, medium or high potential in consultation with staff working with sales. Then, the values were one-hot encoded, which resulted in the addition of the three columns to the feature matrix. In order to see potential relevance of this data, the distribution of the low, medium and high potential leads across the two potential outcomes can be looked at:

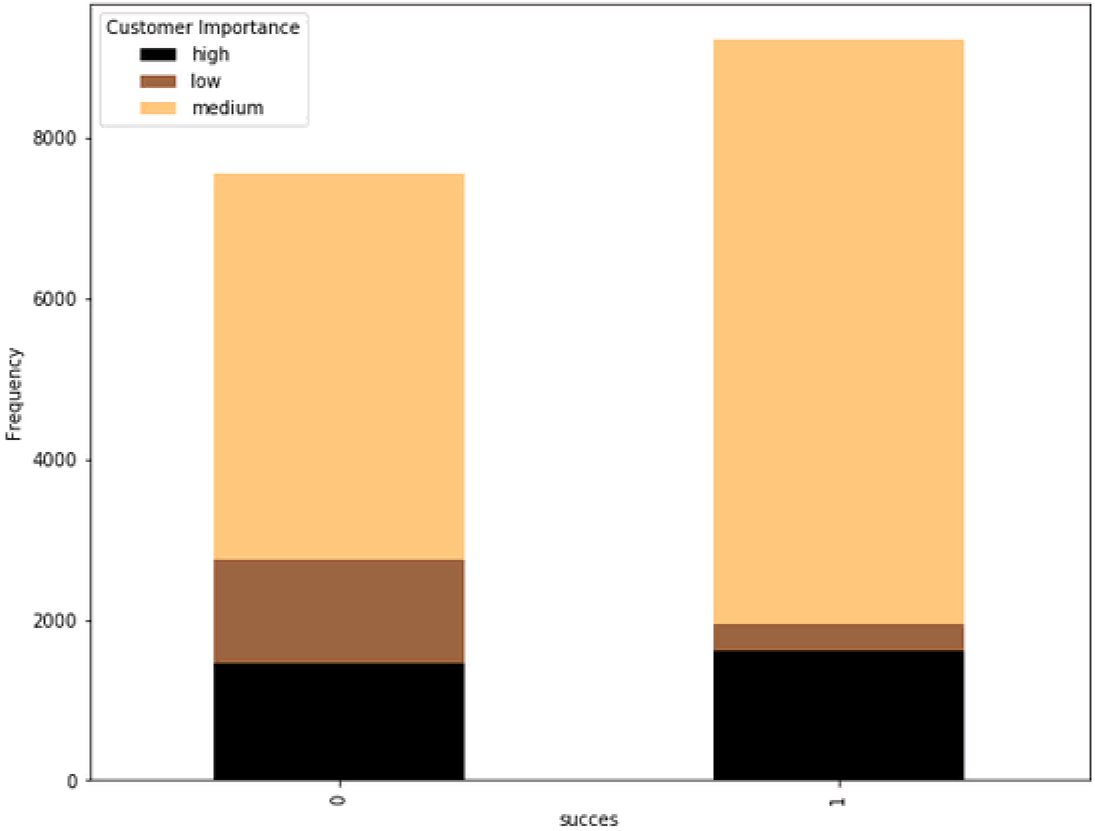


Figure 21: Distribution of customer potential across successful (1) and unsuccessful (0) leads

It can be seen that there are slightly more ‘high’ potential customers in the successful leads and there are more ‘low’ potential customers in the non-successful leads. The performance results and runtimes after the addition of the customer potential can be found in Figure 22 below:

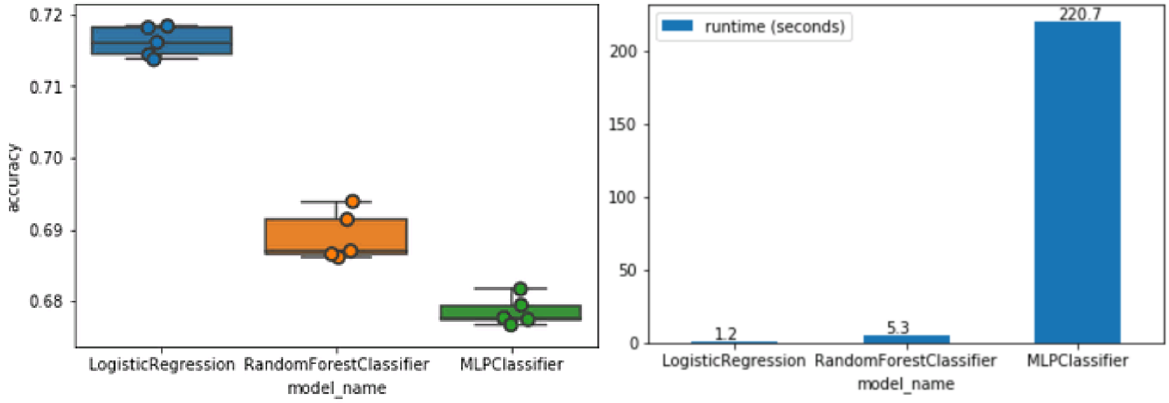


Figure 22: Runtime and performance of the three algorithms after addition of team ids

The addition of this data improved the performance of all three algorithms by about 2%. This is not particularly convincing with regards to keeping these features for the final model. In terms of runtime, while the logistic regression and random forest are basically unchanged, the addition of only three new columns to the feature matrix has resulted in an increase of 16 seconds for the perceptron, which is due to the relatively more complex nature of the calculations performed. The exact results of the evaluation can be found in Table A-4 in Appendix I.

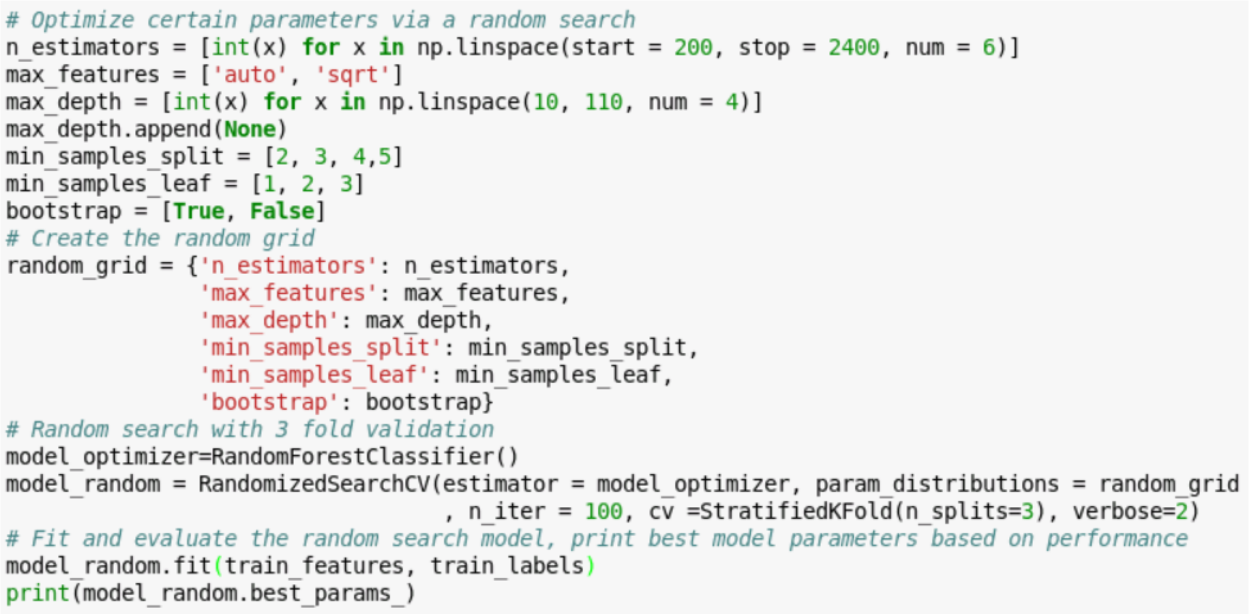
Additionally, a consideration to be made about using this particular variable as a feature. The values of this categorical variable will not be updated unless they are delivered as a lead. This can lead to bias in the evaluation of the leads as the ‘high’ potential customers will always be recommended and other customers which at some point in time were rated lower will be alienated, which in turn means their value for the customer potential is not updated further.

As a result, this feature will not be included in the final model, but can be a consideration for future implementations. Having selected the features and the general model settings, the next step is to perform hyper-parameter tuning per algorithm.

4.3 Model Parameter Tuning & Results

In this section each model will be tuned to find its best setting based on the procedure explained in section 3.3, by using a combination of grid search and random search. It is worth noting that the list of the parameters per algorithm which were tuned using this method, their default values and their final values after tuning can be found in Table A-6 in Appendix I.

The operation can be performed by using the GridSearchCV and RandomizedSearchCV functions of the sk-learn package. First, a grid of relevant parameters must be defined, and then, one of the mentioned functions can be used to perform the search across the grid using a specific algorithm. Example code for performing such a search can be seen in Figure 23 below:



```
# Optimize certain parameters via a random search
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2400, num = 6)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num = 4)]
max_depth.append(None)
min_samples_split = [2, 3, 4,5]
min_samples_leaf = [1, 2, 3]
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
# Random search with 3 fold validation
model_optimizer=RandomForestClassifier()
model_random = RandomizedSearchCV(estimator = model_optimizer, param_distributions = random_grid
                                  , n_iter = 100, cv =StratifiedKFold(n_splits=3), verbose=2)
# Fit and evaluate the random search model, print best model parameters based on performance
model_random.fit(train_features, train_labels)
print(model_random.best_params_)
```

Figure 23: Example of a random search for with a random forest model

The parameters and range of values are defined first, and then the grid is defined as the collection of all the parameters. Afterwards, the model is fit on the data using a certain number of randomly drawn samples (defined by the n_iter argument). The performance of each model setting is evaluated using 3-fold cross validation and eventually, the parameters of the best performing model are printed. If required, these parameters can be manually expanded to create a smaller grid with a more informed starting point, in order to then perform a grid search. The procedure for the grid search is exactly the same as seen in Figure 23 above, except that the number of samples are no longer required (as all possible combinations will be tried), and that the GridSearchCV function will be used instead of RandomizedSearchCV. The performance results of the three models after tuning (with the best parameters) can be seen in Figure 24 below:

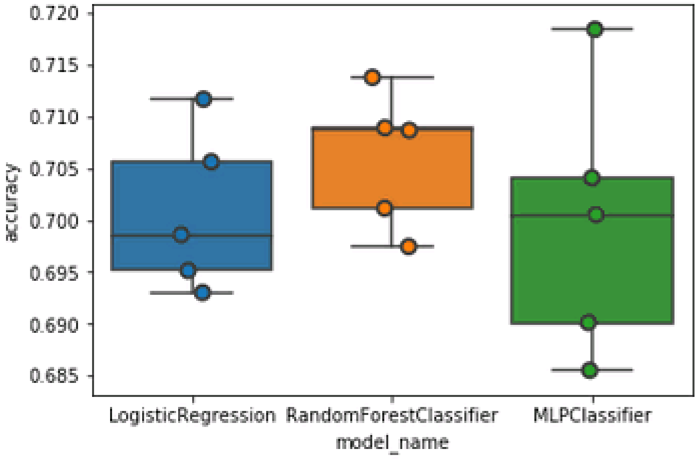


Figure 24: Performance of the three algorithms after parameter tuning

For the logistic regression, given the low fitting times and number of parameters, a grid search for 35 candidates was directly performed. The results of the best setting from the search are almost identical to the default setting. Optimization of the penalty term also helps create a more robust model with regards to overfitting, but as there is no change in the term it suggests that the default value was already optimal.

For the random forest, given the potentially high fitting times, first a random search of a grid of 1440 candidates is performed. Based on probability calculations performed in section 3.3, a random sample of 100 candidates are drawn from the grid. After the random search, the best parameters are selected and manually expanded in order to achieve a much smaller grid of 81 candidates. A grid search is then performed and the best performing parameters are selected for the model. Note that by using a random search first, the total number of candidates for fitting were decreased from 1440 to 181. This operation increases the performance by about 5% and brings it above the performance of the logistic regression. The increase in the number of trees means there are more weak learners in the model which seems to increase the aggregate performance of the algorithm.

For the perceptron, a similar procedure to the random forest is followed, with the method being a random search on a larger grid of followed by a grid search on a smaller grid. By using the random search first, the total number of candidates for fitting were decreased from 1080 to 140. hyperparameter tuning increases the performance by about 6%, bringing it on level terms with the other two algorithms. Overall, after the hyperparameter tuning all three models perform within the same range. With the best hyperparameters chosen per algorithm, in the following section further testing will be performed for validation, based on the results of which the best model will be selected for testing in Chapter 5.

4.4 Model Validation & Best Model Selection

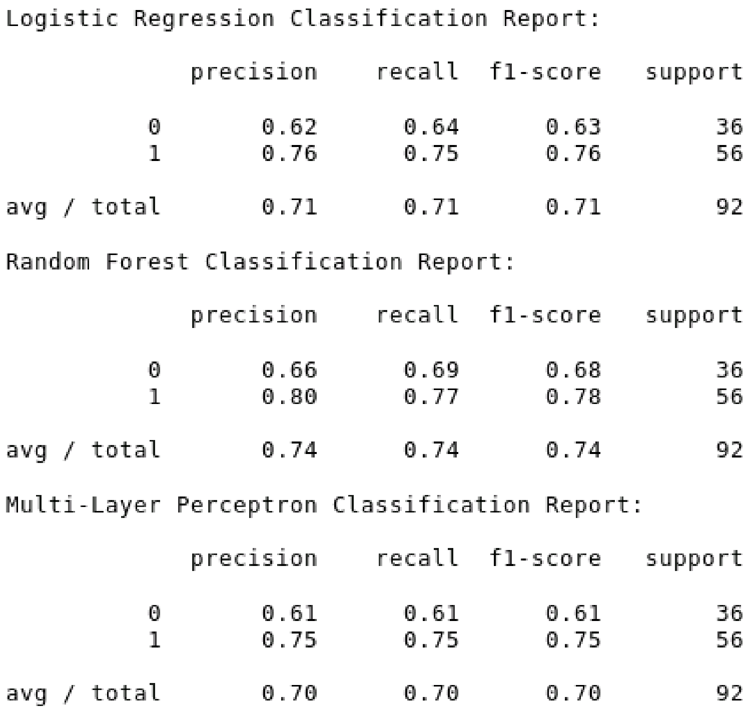
Before selecting a final model for the A/B testing, one more test is performed in order to validate previous performance metrics obtained for all three models. The set-up of the test is:

1- Train: All three models will be trained with labelled lead data up to and including one week before the time of the test. The model settings will be set according to the standards chosen after the model parameter tuning in the previous section.

2- Predict: New leads from the week of the test will be predicted using each model.

3- Evaluate: After the feedback the leads from the week of the test are received from the sales teams, the predictive performance of each model will be evaluated against them. This evaluation is done by calculating the precision, recall and f1-score of each model over the new data-set using a 3-fold cross validation to increase robustness of the results obtained. The best performing model based on this test will be selected for A/B testing against the existing scoring method.

Note that, while this is a valid way of comparing the three models with each other, it does not tell anything yet with regards to the performance of any model compared to the existing method of scoring. This is because the pool of leads which are used as training and new data are already the ones selected by the current scoring. However, given the circumstances of the project and the fact it cannot be afforded that all three models will be tested with live data, this was deemed to be the best option available. The results of this test can be seen in Figure 25 below:



```
Logistic Regression Classification Report:

             precision    recall  f1-score   support

          0       0.62      0.64      0.63        36
          1       0.76      0.75      0.76        56

avg / total       0.71      0.71      0.71        92

Random Forest Classification Report:

             precision    recall  f1-score   support

          0       0.66      0.69      0.68        36
          1       0.80      0.77      0.78        56

avg / total       0.74      0.74      0.74        92

Multi-Layer Perceptron Classification Report:

             precision    recall  f1-score   support

          0       0.61      0.61      0.61        36
          1       0.75      0.75      0.75        56

avg / total       0.70      0.70      0.70        92
```

Figure 25: Performance of the three models with new data from the test week

It can be seen that the model performances with the new data are identical and sometimes surpass the values obtained from the earlier testing. The performance differences are not very conclusive, but in any case, the model with the highest performance will be selected for the live testing. From a business perspective, the most important metric is the ability of the model to identify leads which will be successful correctly, which would be equal to the recall score of label 1. In this regard and also in terms of overall performance, the random forest model is slightly outperforming the other two, and therefore is selected for A/B testing, the setting for and results of which will be explained in the following chapter 5.

5 Testing & Results

In this chapter, first the concept of urgent and non-urgent triggers will be introduced. Following from that, the random forest model will be explored, by looking at the underlying trees and the most important features which affect the decision making, among others. Based on all the previous information analyzed, priors will be presented about the potential results of the testing. Based on the results obtained in the subsequent sections, it'll be discussed whether these priors are supported or challenged in chapter 6.

Additionally, the results of 6 weeks of A/B testing with live data are presented, discussed and evaluated. In the following section, the results are the testing are presented. In the first 4 weeks, both models will work only based on the triggers of the leads. In the last 2 weeks, the team ids from the new data are also extracted as features for the random forest model, in order to see if it makes as much difference with live data as it did during the model design phase.

In the final section, the results of the banker survey will be presented and discussed, with a potential implementation method being proposed. The reason for this is that the survey could not be performed in a timely manner for potential implementation in the scoring before the testing began.

5.1 Urgent & Non-Urgent Leads

As mentioned earlier, one way of increasing performance in call centers has been shown to be by increasing the engagement of the points of contact in the process of their work selection. To look at the effect of the point of contact, it can be analyzed from two directions. First, which kinds of leads are more likely to be picked up by the bankers in each region and second, which kind of leads are more likely to be positively received by the clients in each region. To answer the first question, a survey is performed, the results of which will be presented further in section 5.4 in this chapter. To answer the 2nd question, the performance of the teams across different triggers was investigated in Figure 19 in the previous chapter (section 4.2.1) which showed what triggers were the most successful in each region when a call was made.

Based on that figure and after consultation with the bankers and staff at the department, a list of 26 triggers were deemed to be always important. Most of these can also be found in the top performing triggers in Figure 12 in section 4.1 (for the full list, please refer to Table A-7 in Appendix I). It was decided that every lead which contains at least one or more of these triggers is 'urgent' and will always be sent out. Therefore, the models have to pick the best of the 'non-urgent' leads based on triggers which are not obviously good and the performance of both models will be evaluated on this subset. Having clarified this difference, in the following section the structure of the random forest will be examined.

5.2 Exploring the Random Forest

In this section the structure of the forest and its underlying trees will be examined. Based on the results of the hyperparameter tuning, the forest was made with 2000 trees and a maximum depth of 120 per tree. Upon investigation of the tree depths, it seems that all trees have reached the maximum depth, which may suggest that the depth could have been increased, at the cost of higher computational resources and time. In order to further investigate this, the percentage of pure leaf nodes can be looked at. This means the distribution of the ratio of pure decision nodes (containing data only from one label) over the total number of leaf nodes per tree over the forest. In simpler terms, this equates to the number of nodes at which a classification decision can be made with absolute certainty. This can be seen in Figure 26 below:

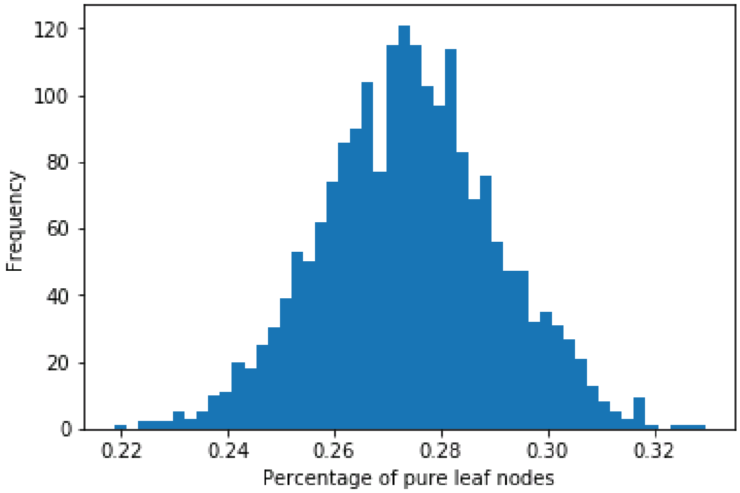


Figure 26: Distribution of the percentage of pure leaf nodes over the trees of the forest

The distribution is relatively normal, centered around 27%. This suggests that perhaps the current depth of the tree is not particularly low. Whether a higher ratio of pure nodes could be achieved with deeper trees is arguable, but based on the fact that during the parameter optimization the option trees with higher depth was available and was not selected, the possibility is not high. Additionally, the total number of nodes per tree can be looked at. The distribution of this measure over the 2000 trees can be found in Figure 27 below:

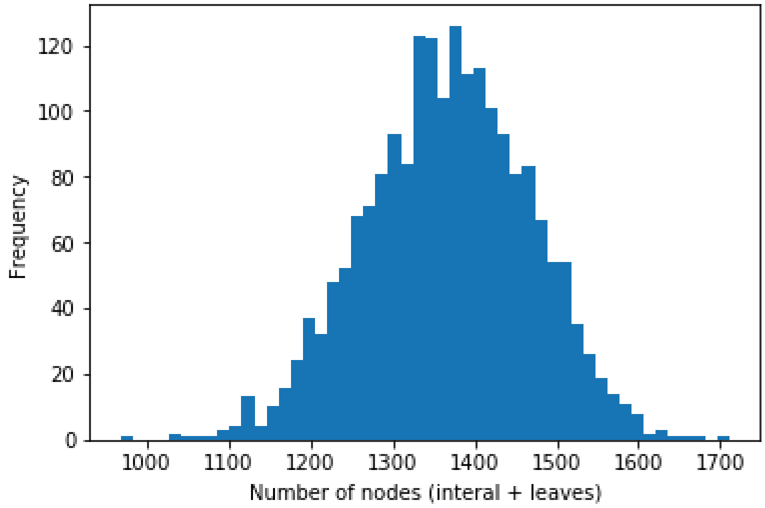
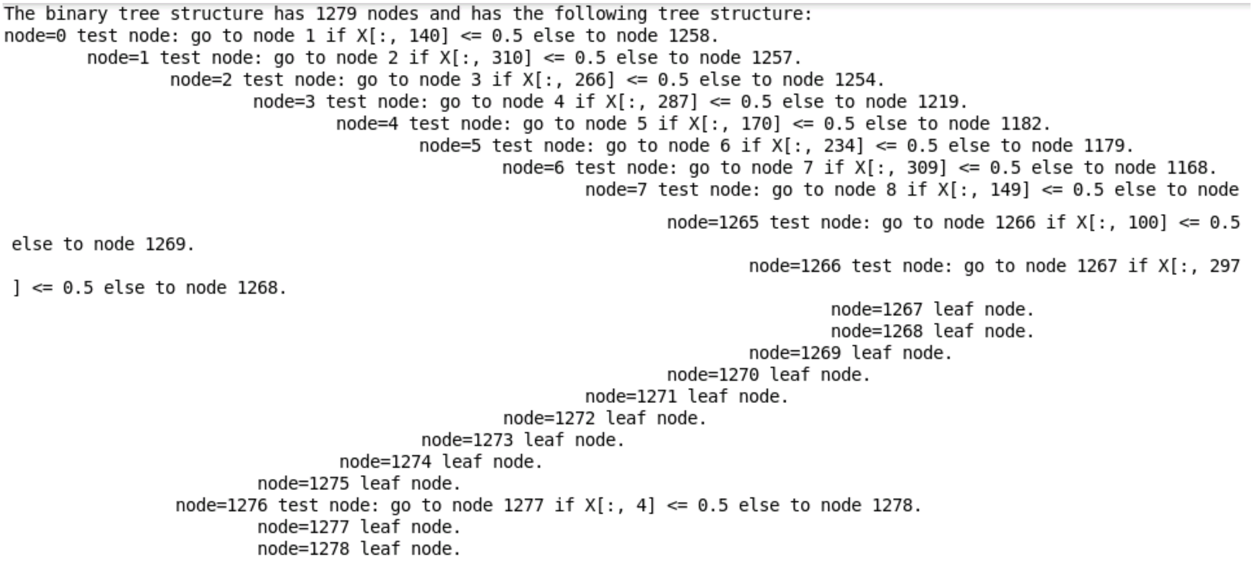


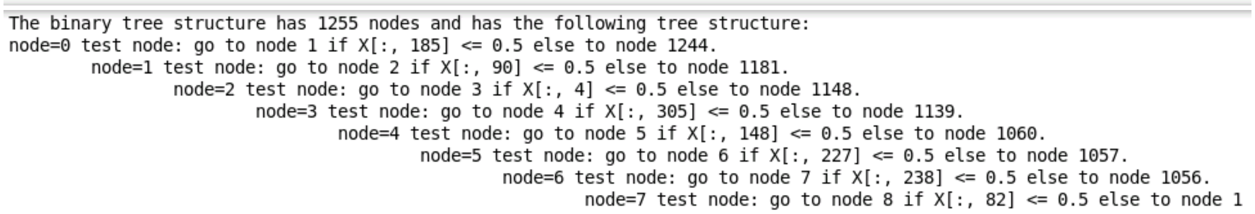
Figure 27: Distribution of the total number of nodes over the trees of the forest

The distribution seems relatively normal, centered around the 1400 mark. This shows the majority of trees are over 1000 nodes big, which would make any visualization of the trees unreadable. However, different trees can be looked at in certain depths in order to see if they value different features more or less. In order to do this, the decision path of each tree can be exported in text form. Two randomly selected trees from the forest will be compared in this way In Figure 28 below, the start and end of such a text file for two trees (1st tree on top and 2nd tree on the bottom) can be seen:

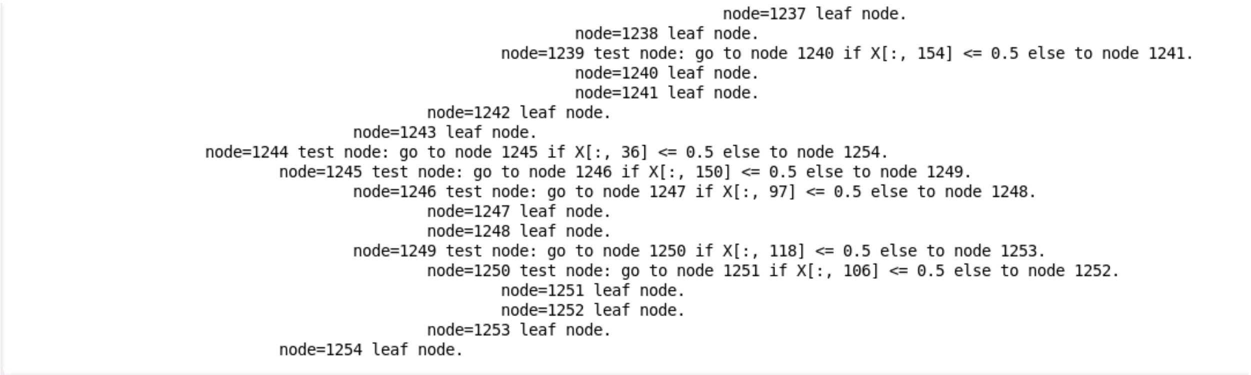


```
The binary tree structure has 1279 nodes and has the following tree structure:
node=0 test node: go to node 1 if X[:, 140] <= 0.5 else to node 1258.
        node=1 test node: go to node 2 if X[:, 310] <= 0.5 else to node 1257.
                node=2 test node: go to node 3 if X[:, 266] <= 0.5 else to node 1254.
                        node=3 test node: go to node 4 if X[:, 287] <= 0.5 else to node 1219.
                                node=4 test node: go to node 5 if X[:, 170] <= 0.5 else to node 1182.
                                        node=5 test node: go to node 6 if X[:, 234] <= 0.5 else to node 1179.
                                                node=6 test node: go to node 7 if X[:, 309] <= 0.5 else to node 1168.
                                                        node=7 test node: go to node 8 if X[:, 149] <= 0.5 else to node

                                                                node=1265 test node: go to node 1266 if X[:, 100] <= 0.5
 else to node 1269.
                                                                        node=1266 test node: go to node 1267 if X[:, 297
 ] <= 0.5 else to node 1268.
                                                                                node=1267 leaf node.
                                                                                node=1268 leaf node.
                                                                        node=1269 leaf node.
                                                                node=1270 leaf node.
                                                        node=1271 leaf node.
                                                node=1272 leaf node.
                                        node=1273 leaf node.
                                node=1274 leaf node.
                        node=1275 leaf node.
                node=1276 test node: go to node 1277 if X[:, 4] <= 0.5 else to node 1278.
                        node=1277 leaf node.
                        node=1278 leaf node.
```



```
The binary tree structure has 1255 nodes and has the following tree structure:
node=0 test node: go to node 1 if X[:, 185] <= 0.5 else to node 1244.
        node=1 test node: go to node 2 if X[:, 90] <= 0.5 else to node 1181.
                node=2 test node: go to node 3 if X[:, 4] <= 0.5 else to node 1148.
                        node=3 test node: go to node 4 if X[:, 305] <= 0.5 else to node 1139.
                                node=4 test node: go to node 5 if X[:, 148] <= 0.5 else to node 1060.
                                        node=5 test node: go to node 6 if X[:, 227] <= 0.5 else to node 1057.
                                                node=6 test node: go to node 7 if X[:, 238] <= 0.5 else to node 1056.
                                                        node=7 test node: go to node 8 if X[:, 82] <= 0.5 else to node 1
```



```
                                                        node=1237 leaf node.
                                        node=1238 leaf node.
                                node=1239 test node: go to node 1240 if X[:, 154] <= 0.5 else to node 1241.
                                        node=1240 leaf node.
                                        node=1241 leaf node.
                        node=1242 leaf node.
                node=1243 leaf node.
node=1244 test node: go to node 1245 if X[:, 36] <= 0.5 else to node 1254.
        node=1245 test node: go to node 1246 if X[:, 150] <= 0.5 else to node 1249.
                node=1246 test node: go to node 1247 if X[:, 97] <= 0.5 else to node 1248.
                        node=1247 leaf node.
                        node=1248 leaf node.
                node=1249 test node: go to node 1250 if X[:, 118] <= 0.5 else to node 1253.
                        node=1250 test node: go to node 1251 if X[:, 106] <= 0.5 else to node 1252.
                                node=1251 leaf node.
                                node=1252 leaf node.
                        node=1253 leaf node.
        node=1254 leaf node.
```

Figure 28: Start and end of the decision path of two trees of the forest

In these figures X represents the feature matrix on which the model was trained and the number in X[:, n] represents the encoded feature which was at nth column of the feature matrix. It can be seen that the trees have different structure and additionally, value different features closer to their roots. This shows that different trees in the forest can learn different things from the data, and is the basis upon which the ensemble approach is supposed to be superior to a single decision tree. By aggregating the decision-making from multiple decision-trees (in this case 2000), the probability of bias is reduced and a more robust decision can be made, based on a large amount of information.

In order to better interpret the results of the model, analysis of the features contributing to the decision making is useful. In order to do so, some form of feature importance needs to be evaluated. This is done to gain an understanding of how the algorithm works and the decision making of the algorithm, in order to facilitate the governance of the algorithmic decision making process, as explained earlier in chapters 1 and 2. On the model level, this can be done by using the mean Gini impurity decrease. This is defined as the average of a features' total decrease in node impurity (for the formal definition of the Gini impurity please refer to section 3.1.2), weighted by the proportion of samples reaching that node in each individual decision tree making up the random forest. This is in essence a measure of how important a feature is for estimating the probability value of the target label across all of the trees that make up the forest. A higher value of mean Gini decrease means a higher feature importance. For the random forest model, the calculation yields the following results seen in Figure 29 below:

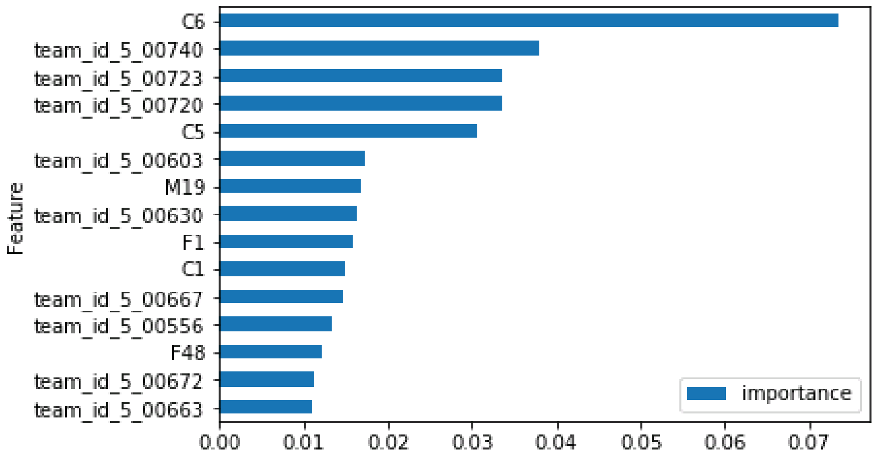


Figure 29: Overall feature importance of the model

It can be seen the majority of the top 15 decision making factors of the model are to which teams the leads are being sent to. This further justifies the addition of the team ids as features and is consistent with the expectation of prior 3 as defined earlier. Additionally, the three compliance triggers with the highest conversion (C6, C5 & C1) are also present. This is a bit worrying, as with the way the testing is done only non-urgent leads are scored by the model and the significance of these urgent triggers could lead to a negative bias towards the non-urgent leads. So far, an aggregated view of the decision making of the model is obtained. In order to go deeper, single decisions must be looked at and dissected. To do so, a method of decomposition is used, where the decision path of the probability calculation is dissected step by step, and the final probability value is decomposed into the sum a bias and the contributions of each feature dominant in each decision node located on the decision path (Saabas, 2014). Then, the prediction function is in the form of:

$$5.2.1 \quad p(x) = bias + \sum_{k=1}^{K} contrib(x, k)$$

Where the bias for each label is the label ratio in the training dataset (value at the root of the tree), K is the number of features and contrib(x,k) is the probabilistic contribution from the kth feature of the feature matrix x. In order to gain some insight into single decision, one of the non-urgent leads with the highest probability of success as calculated by the model from week 4 is selected. This lead has the following features as seen in Table 3 below:

Table 3: Features of the selected lead

Feature	**Type**
F2	Trigger
F18	Trigger
F6	Trigger
F31	Trigger

P78	Trigger
team_id_5_00640	Team id

Then, the six highest marginal contributions to the probability of success of the lead can be investigated. These values can be seen in Figure 30 below:

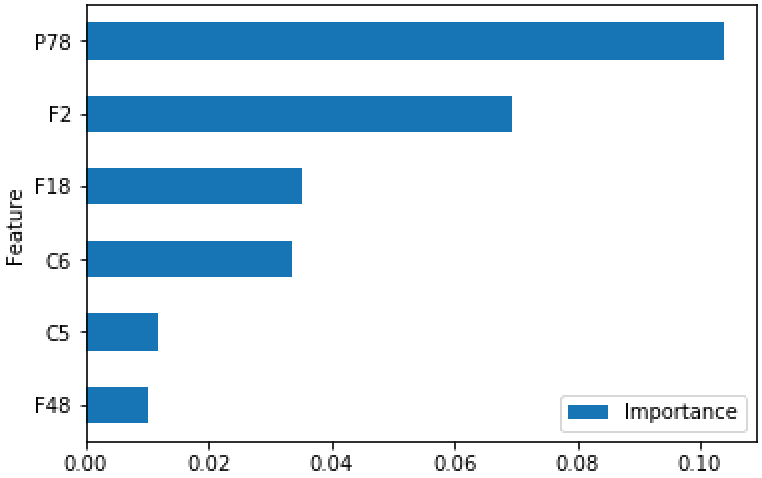


Figure 30: Six highest marginal contributions to the final probability calculation for the lead

While the top 3 contributions are from the triggers that are in the lead, it can be seen that the other 3 contributions are from the absence of certain high conversion urgent triggers. This is not ideal for the probability calculation, as it may overshadow other present triggers in the lead, for example the team id. As explained earlier, the reason this has happened is due to the fact the model was trained with all the data, urgent and non-urgent leads alike, and is now only being used to select the best non-urgent leads. The solution then could be to re-train the model only with non-urgent leads. The argument for this would be that the judgment of the model will not be clouded by practically pointless high conversion urgent triggers, and the argument against it would be based on the potential information loss from the other triggers inside the excluded urgent leads in the training set.

Partially based on this analysis, the following two changes are made for the model which will be used for the final 2 weeks of A/B testing:

1- The random forest model was re-trained only with non-urgent leads from the past year

2- The feature extraction process was modified to include the team ids for all non-urgent leads as features

After these changes, the structure of the new forest can be looked at. The distribution of the depths of the trees in the forest can be seen in Figure 31 below:

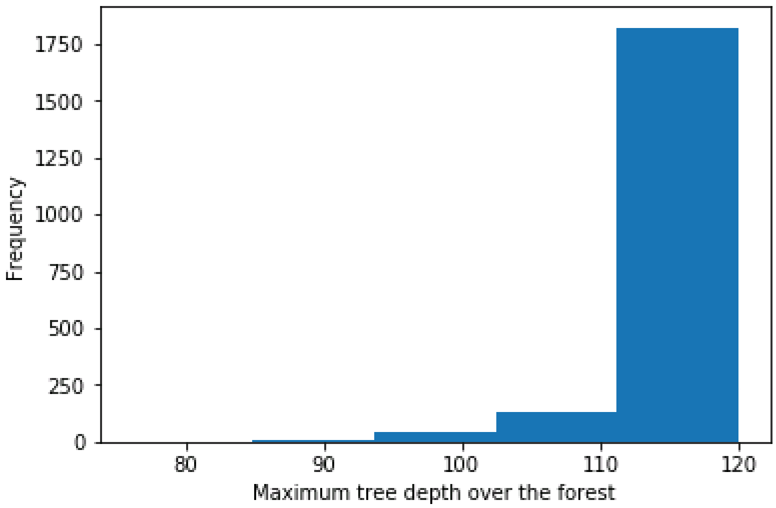


Figure 31: Distribution of tree depth over the forest

It can be seen that unlike before the changes were made, not all trees reached the maximum depth. This can be attributed to the reduced number of features as a result of removing the urgent triggers from the training feature matrix. Still, 87.5% (1750 out 2000) of the trees reach the maximum depth. Furthermore, The distributions of both percentage of pure leaf nodes as well as the number of total nodes of the forest can be seen in Figure 32 below:

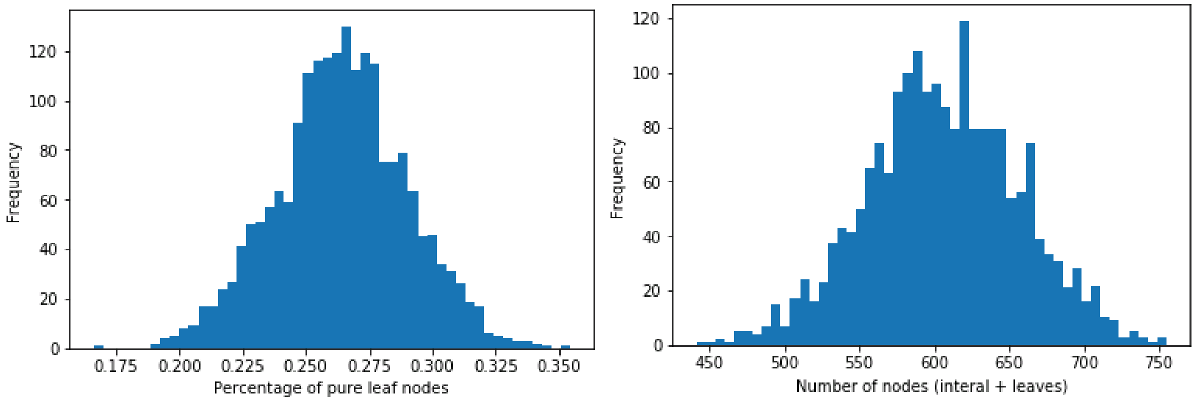


Figure 32: Distribution of pure leaf nodes (left) and number of nodes (right) of the trees over the forest

It can be seen the number of trees with above 30% pure leaf nodes has slightly increased (especially around the 32% mark), this can be as a result of the new information about the teams because of the addition of the team ids. However, the lower tail of the distribution also seems to be slightly more populated. This can suggest some information loss as a result of the removal of the urgent leads. Furthermore, it can be seen that the total number of nodes of the trees have significantly reduced, by almost 1000 nodes per tree. This further supports the idea that decisions are being arrived at more quickly and with better accuracy. Additionally, the feature importance of the modified model can be looked at. Again, first the overall feature importance is investigated, as seen in Figure 33 below:

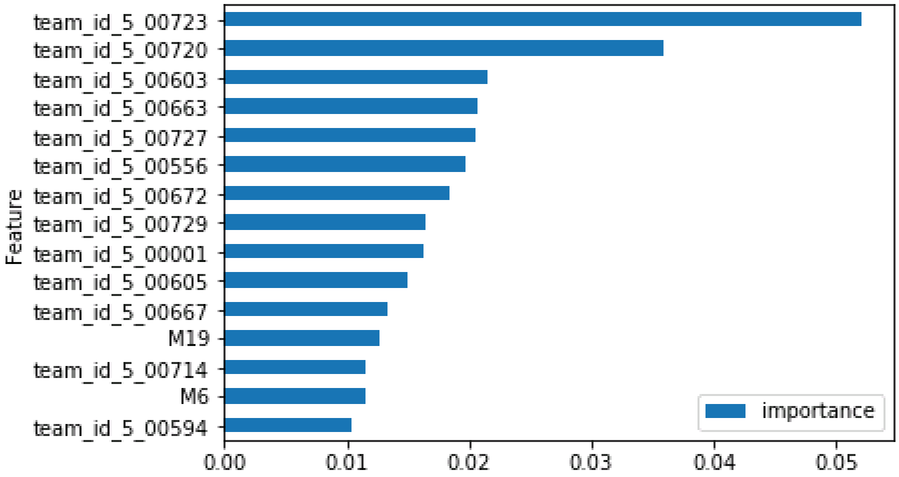


Figure 33: Overall feature importance of the modified model

The most prominent change compared to before is the fact now out of the 15 most prevalent decision contributors of the model, only two are triggers and the rest are the teams. Additionally, it can be seen that urgent triggers are no longer a factor in the decision making of the model, which was the purpose behind re-training the model. Following from these, a single decision can again be examined based on the decomposition process explained earlier. The particular lead in question this time has the following triggers, as seen in Table 4 below:

Table 4: Features of the selected lead

Feature	Type
M6	Trigger
P29	Trigger
F31	Trigger
C7	Trigger
team_id_5_00001	Team id

Then, the five highest marginal contributions to the probability of success of the lead can be investigated. These values can be seen in Figure 34 below:

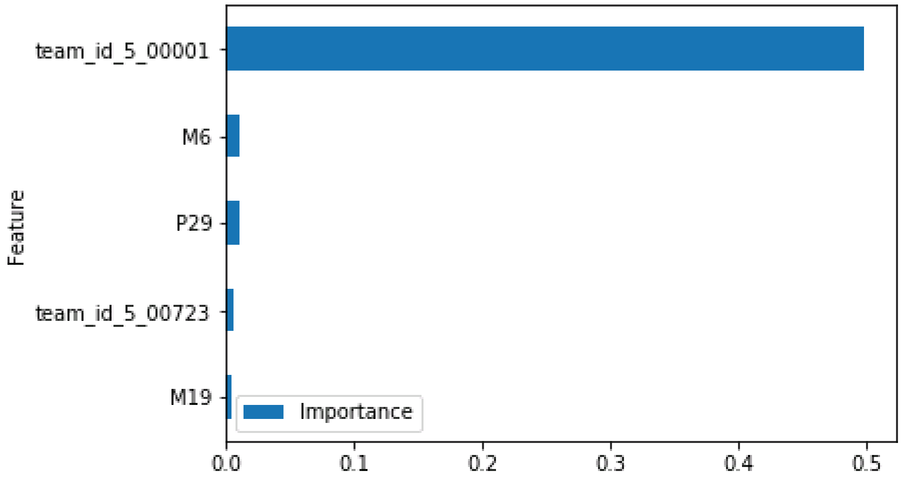


Figure 34: Five highest marginal contributions to the final probability calculation for the lead

It can be seen that the team that is receiving the lead is by far the most important factor in the decision making of the model, followed by 2 of the non-urgent triggers present in the lead. The other two contributions seem to be irrelevant, but their contribution values are also very small. This suggests an overall improvement compared to before the changes were made on the model settings. It seems that now the model considers both the non-urgent triggers and the teams ids as important decision making factors, which is in line with what was expected. It remains to be seen whether similar results will be obtained in the live testing with new data. This will be investigated in the discussion and evaluation section further below.

5.2.1 Priors

Before presenting the A/B testing results and discussion, in this section some prior beliefs will be presented. These are based on information presented either in earlier chapters or in previous sections of this chapter, as well as general understanding of the environment and the case at hand:

Prior 1: The urgent leads will have a higher conversion compared to non-urgent leads, as they contain some of the best performing triggers as seen in Figure 2 and some triggers preferred by the bankers.

Prior 2: Given the marketing-oriented nature of the private banking department, product (P) and financial (F) related triggers will be picked more frequently than others.

Prior 3: After addition of team ids as features in week 4, in the random forest model the team ids will play an important role in the decision making and will increase the performance of the model, as seen in the testing earlier in this section.

Prior 4: Based on the survey results, better performing regions have higher correspondence between their preferred trigger types and ones that they perform well on, which leads to a more motivated effort on the side of the bankers to have successful calls.

Prior 5: Based on the survey results, the bankers will show a higher preference for urgent triggers than non-urgent, which would be consistent with the earlier agreement to always send all urgent leads.

Exploring priors 1 and 5 based on the results can help quantify the merit in the selection method used for the testing. Exploring priors 2, 3 and 4 can help in gaining insight about triggers and how they might affect lead performance, thus helping to answer research SQs 3 and 4. Furthermore, prior 4 can help identify potential for implementation of banker preference in the lead scoring and selection procedure. After presenting all the results in the following sections of this chapter, in chapter 6 it will be discussed whether these priors are supported or challenged by the results.

5.3 A/B Testing

In this section in the first part, the settings of test are explained. In the second part, the results of the testing over 6 weeks are presented and in the final part, it is aimed to discuss and evaluate the results by investigating why and how they were achieved.

5.3.1 A/B Test Setting

Following from chapter 4 where the best model was selected, now it must be tested against the existing scoring method in order to evaluate the performance for sending out leads across the country. The test is done by sending out half the number of leads which are supposed to be sent out using the current scoring method and the other half by the probability of success calculated by the random forest model.

To ensure geographical factors or team performance have minimal effect on the results, a double sort on performance and region was done on the list of teams, and then each team in the sorted listed was assigned to either the control group (0) or the treatment group (1). The non-urgent leads belonging to the control group will be scored using the existing model and the non-urgent leads belonging to the treatment group will be selected by the random forest model. On top of these leads, all urgent leads for both groups will be sent out. A systematic representation of the setup of the testing can be found in Figure 35 below:

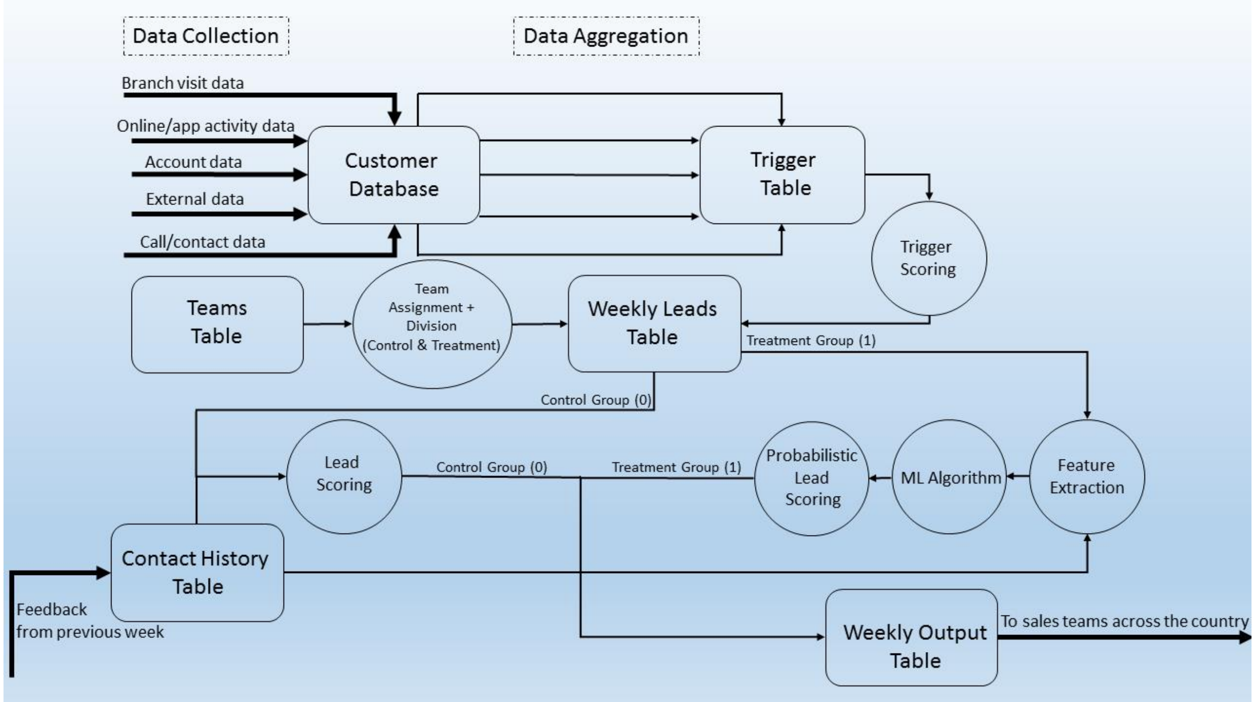


Figure 35: Systematic overview of the A/B testing

The teams in the team table have been assigned a treatment group (0 or 1) based on performance and location, explained earlier. Then, once the leads are assigned to their teams in the weekly lead table, they are also assigned to their treatment group. The leads of treatment group 0 will continue as normal using the existing method, whereas the leads of treatment group 1 will be processed for feature extraction and fed into the random forest model, which then assigns each lead a probability of success. In the end, both treatments groups are added together in the final output table to be sent out.

Once feedback is received, the performance of the models can then be evaluated based on which treatment group the lead belonged to. From an operational perspective, the process of using the random forest model can be seen in Figure 36 below:

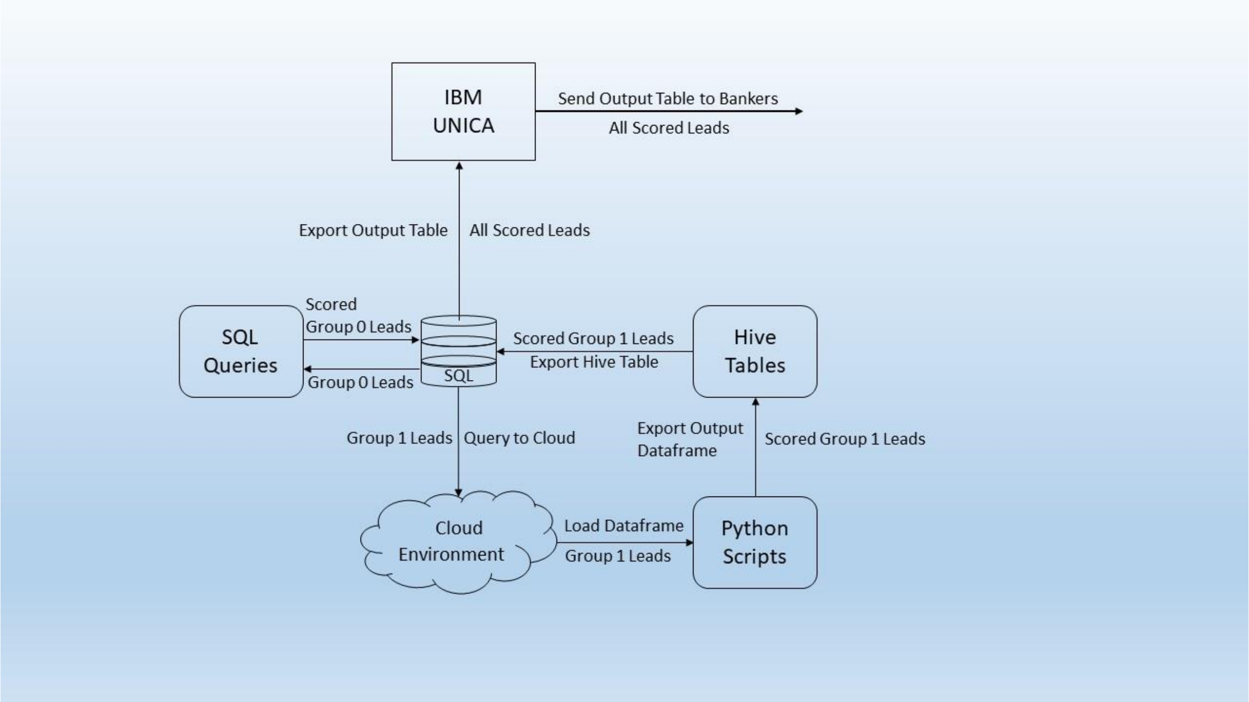


Figure 36: Operational overview of the A/B testing

The integration of the data analytics methods into the existing infrastructure was made possible by using several different environments in combination. The leads of treatment group 0 are directly scored using SQL queries (with the existing scoring method) and are exported to a table in the SQL database and ready to be read into unica. The leads of treatment group 1 are first exported to a cloud environment where Python access is available. There, they are loaded as pandas dataframes and processed using Python scripts. The final output dataframe with the scored leads is then written to a Hive environment as a table. Subsequently, the Hive table is exported back to the main SQL database and added to the table containing the scored treatment group 0 leads. In the final step, the table of the scored leads is read into unica, where they can be sent out to the bankers. Having clarified the testing procedure, in the following section the results are presented.

5.3.2 Results

In this section the results of the A/B testing are discussed. The testing can be divided into two parts. As explained, in the first 4 weeks both models will operate on the same set of features, which is all of the triggers available. In the final 2 weeks (weeks 5 and 6), the random forest model is retrained using only non-urgent triggers and the team ids from the data are also extracted as features for the model.

For comparison of the performance of the models alone, the metrics on the non-urgent subset must be considered. Nevertheless, it is interesting to look at the conversion of the urgent leads as well, in order to see if the assumption that leads containing these triggers will have a higher success is just. The overall conversion of urgent and non-urgent leads (regardless of which model had selected them) over 4 weeks of testing can be seen in Figure 37 below:

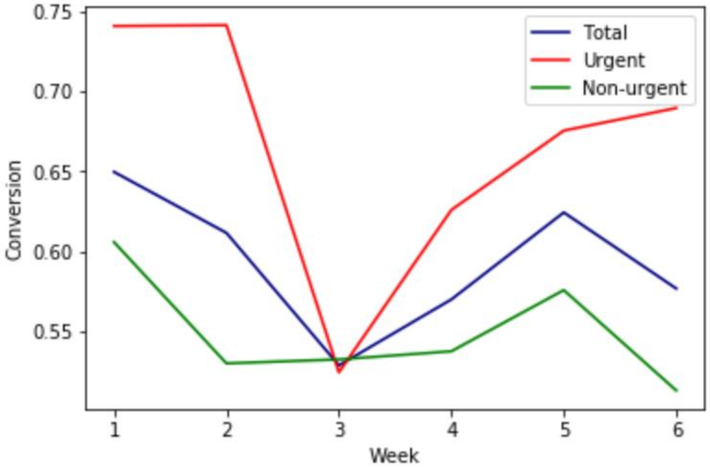


Figure 37: Urgent vs. non-urgent leads conversion

It can be seen that while in the first two weeks the urgent leads have a significantly higher conversion, there is a large dip in the third week, which points to perhaps the urgent triggers not necessarily meriting the high selection priority that they are given due to inconsistency. From the fourth week onwards, there is again a considerable difference between the overall conversion of the urgent leads compared to non-urgent. In order to explain this difference, it would be interesting to see if the urgent leads actually picked up more by the bankers in the first place. To do so, the engagement rate (being defined as the ratio of the number of leads sent and the number of leads for which feedback is received in each week) of urgent and non-urgent leads must be looked at, which can be seen in Figure 38 below:

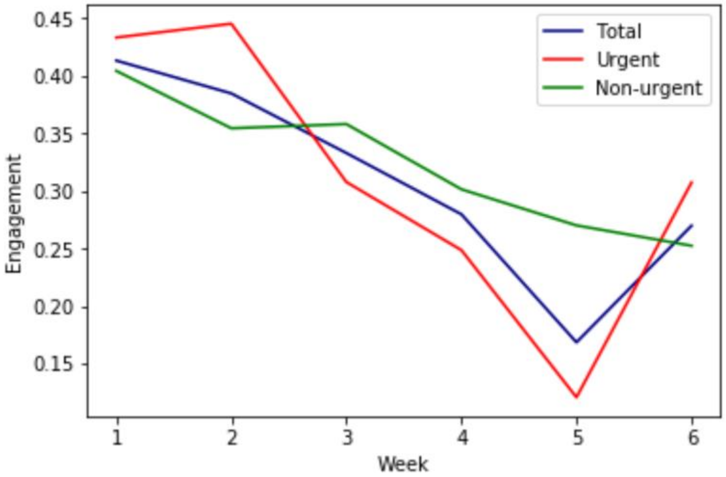


Figure 38: Urgent vs. non-urgent lead engagement

The engagement follows a similar pattern to the conversion, with the engagement of the urgent leads taking a sharp dip in from the third week onwards. This may also explain the dip in the third week, simply because less of urgent leads were contacted in the first place. The key insight from this figure is the fact that less than half of leads being sent out are getting feedback. This causes a large data leakage in the feedback loop, based on which the majority of data every week is not being labelled for future learning. This can be a clear indication that too many leads are being sent out. The reason for this overselection could be due to the way the final output table is made, in which the total number of leads the bankers asked for are selected from the non-urgent pool, and all of the urgent leads for the week are added on top, which usually doubles the total number of leads sent out. This overload of work may reduce concentration on the leads that are being contacted, while not providing much benefits as for the most part both types of leads are being picked up at the same rate.

Having established the difference in results between urgent and non-urgent leads, next the two models can be compared. From this point onwards, all metrics are calculated for the non-urgent leads subset, as this was the subset on which the two different scoring methods were used for lead selection. The results of the testing can be looked at in terms of conversion. Conversion is defined as the ratio of the number of non-urgent leads for which positive feedback is received over the total number of non-urgent leads for which feedback is received. These results can be seen over the 6 weeks in Figure 39 below:

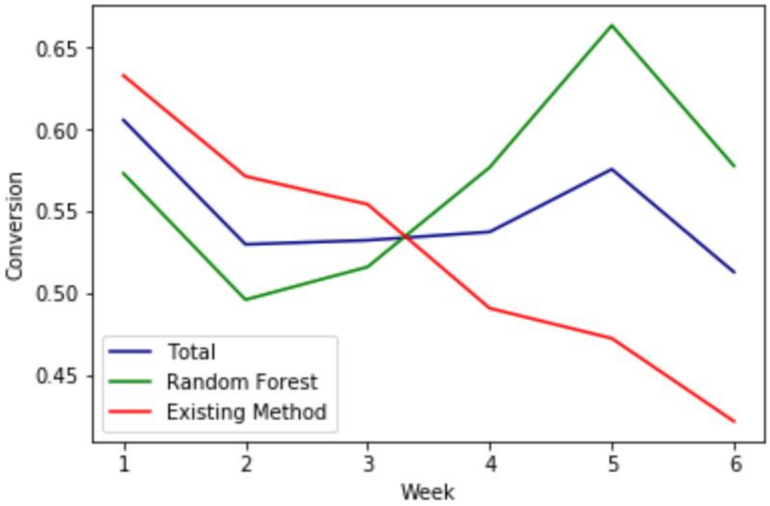


Figure 39: Random forest vs. existing method model performance

It is worth noting that to some extent, the conversion of each week is limited by the quality of the leads and this is very dynamic due to the fact the triggers making up the leads are generated daily. It can be seen that the existing method starts better but slowly drops while the random forest starts worse off but seems to be getting with more learning by the 4th week. In the 5th, which is the first week in which the random forest is modified with team ids and a smaller feature set (only non-urgent triggers), there is significant gain in conversion which peaks at 66%. The following week there is a small dip, but in any case in the last 3 weeks of testing the random forest model is outperforming the existing method considerably. Aside from overall conversion, both positive and negative feedback each have two levels in terms of business value (or loss). These are mapped to four outcomes: good (highest business value), positive, negative, bad (no business value). The good outcome has by far the highest business value. It is then interesting to see the outcome distribution of the two models. These results can be seen over the 6 weeks in Figure 40 below:

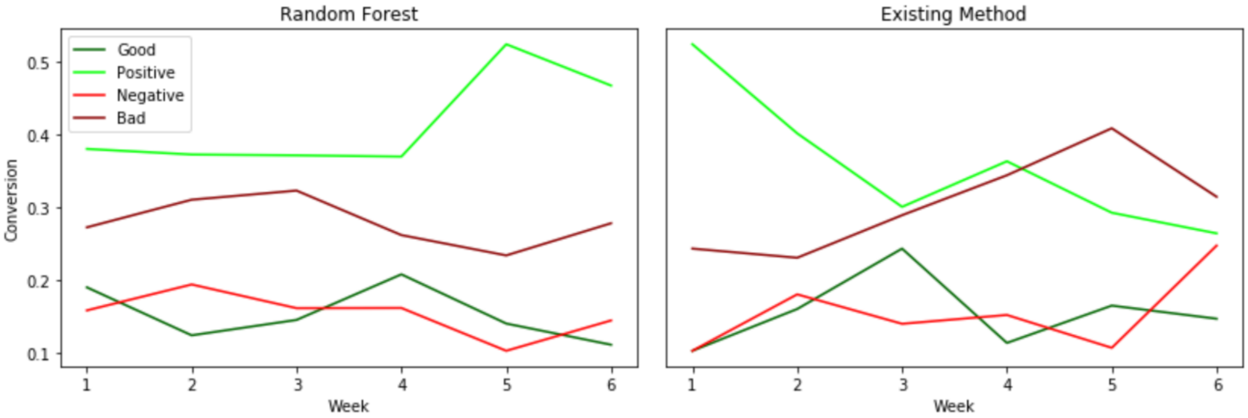


Figure 40: Random forest vs. existing method outcome distribution

It can be seen that the random forest model is more consistently providing a reasonable number of leads with positive outcomes every week. Again, in the last two weeks the modifications have a considerably positive effect on the number of leads with positive outcomes for the random forest model. In terms of good and negative leads, both modes seem to be fluctuating between the 10-20% mark. As these are both results of calls being made, it is reasonable to assume no matter what lead is sent out, there are external factors affecting the exact outcome. The bad leads, however, represent no calls being made. For both models, the bad leads are the second highest frequency outcome. While there is no particular pattern in the existing method, the random forest seems to be gradually decreasing in the percentage of bad leads, except for a slight rise between weeks 5 & 6. Overall, the random forest seems to be getting better as the weeks go by, whereas there is no particular pattern in the performance of the existing method of scoring.

The next topic to examine is whether the random forest model has actually learned anything. In order to assess this, for each week, the conversion of the leads selected by the random forest model are examined at three separate probability thresholds. As the model was optimized with a probability threshold of 0.5 (that is to say the model will predict a positive outcome when it is >50% sure the outcome will be positive), thresholds of >40%, >50% and >60% will be looked at. The expectation is that with the increasing probability thresholds, the conversion must also increase. These results can be seen in Figure 41 below:

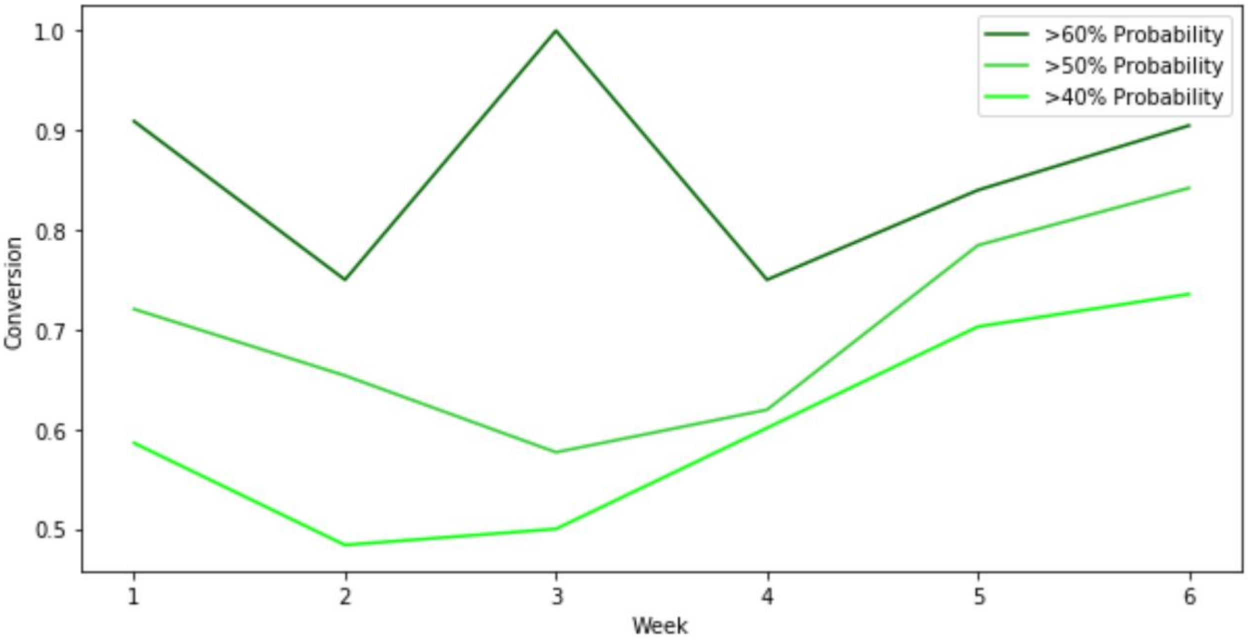


Figure 41: Random forest lead conversions at different probability thresholds

It can be seen that the leads with higher probability of success as calculated by the model are indeed more successful. Especially from weeks 4 to 6, considering the changes made in the last two weeks, the model seems to be getting better at predicting successfully at all three probability thresholds. Another measure to compare is the conversion at the >50% threshold. This was the threshold at which the model was made, and at the last testing before starting these 6 weeks the accuracy of the model was about 74%, as shown in Figure 25 in section 4.4. With live testing, it can be seen that while there is a dip in performance in the first few weeks, but after week 4, the 74% performance metric is reached and even surpassed. This is particularly encouraging with regards to the robustness of the model performance, considering the dynamic nature of the live data testing.

Having established that the model has indeed learned from the data, the next step would be to try and see what exactly has been learned. In the following section, a more in depth into the workings of the two scoring methods is presented, in order to better interpret and understand the results obtained during the testing.

5.3.3 Discussion and Evaluation

In order to better understand the results, it is interesting to see what kinds of triggers are the models sending out. In Figure 42 below, venn diagrams of the top 10 most frequent (non-urgent) triggers selected by both the random forest model and the existing scoring method can be seen:

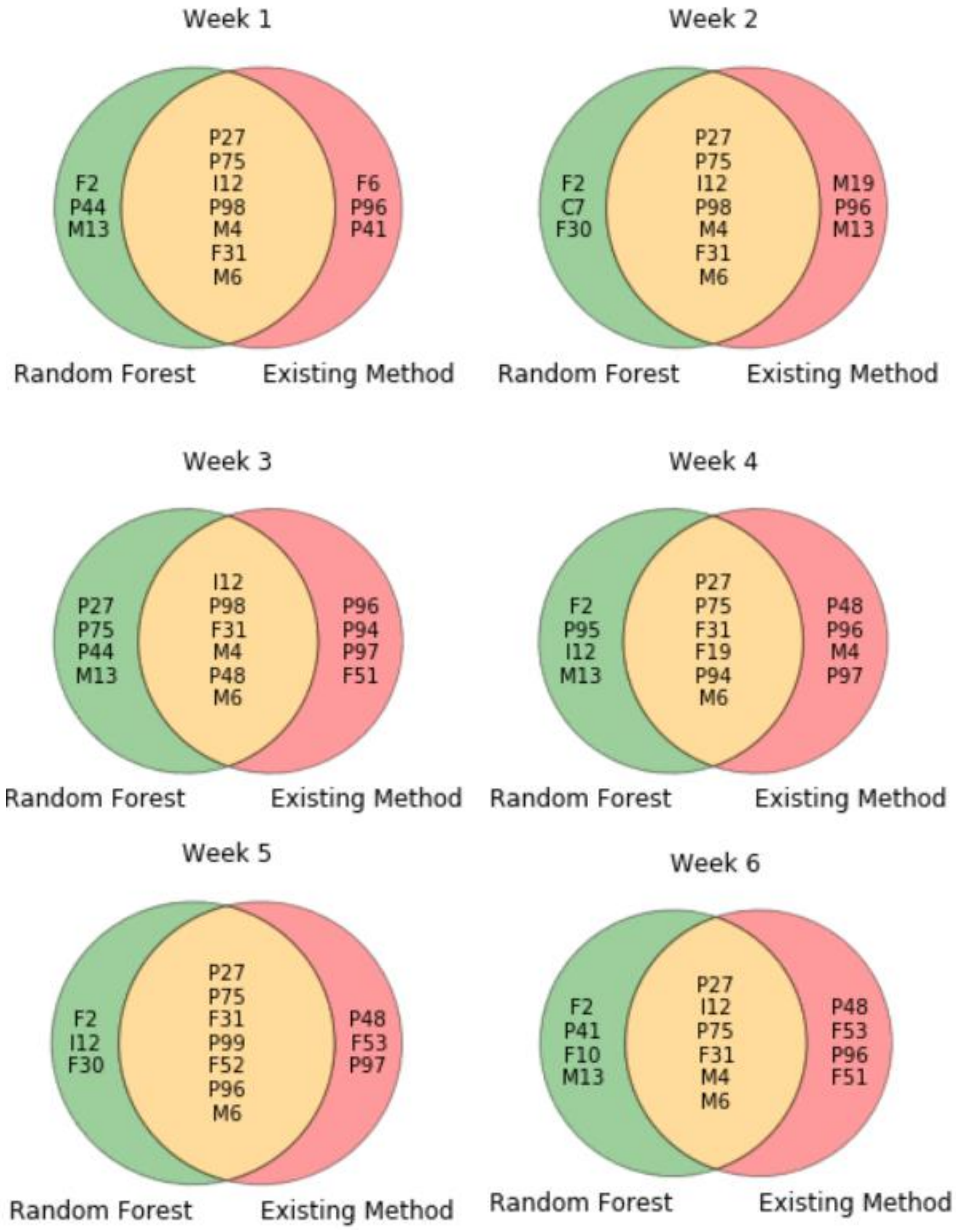
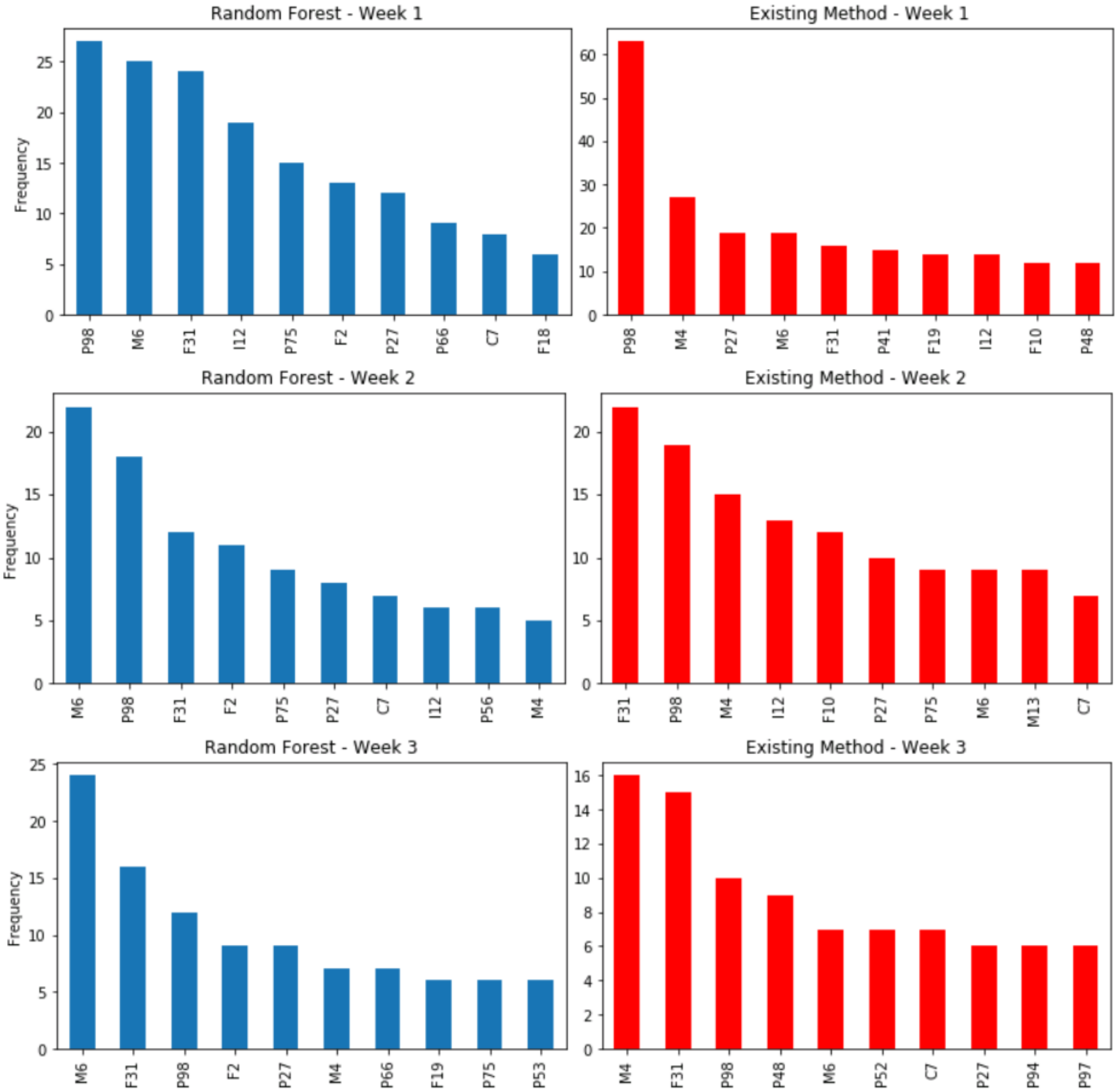


Figure 42: Top 10 most selected triggers for the two models across 6 weeks of testing

In terms of the selection of the triggers, both models seem to be selecting a balance of product, financial and service triggers. In the first 4 weeks, the main difference seems to be in 3 to 4 triggers. The increased difference in selection in weeks 3 & 4 could explain the improvement in the performance of the random forest model. Both models consistently rank F31 and M6 as two of the most important triggers. This in a way is good as it emphasises on the importance of solid evidence of financial capital availability (due to the F31 trigger) as well as keeping focus on customers who have not been contacted for a while, regardless of other factors (due to the M6 trigger) in order to avoid a self-fulfilling prophecy scenario. It would be interesting to investigate whether certain triggers in particular contribute to the success of either model. The top 10 most frequent leads appearing successful leads per model over the 6 weeks can be seen in Figure 43 below:


Random Forest - Week 1

Existing Method - Week 1

Random Forest - Week 2

Existing Method - Week 2

Random Forest - Week 3

Existing Method - Week 3

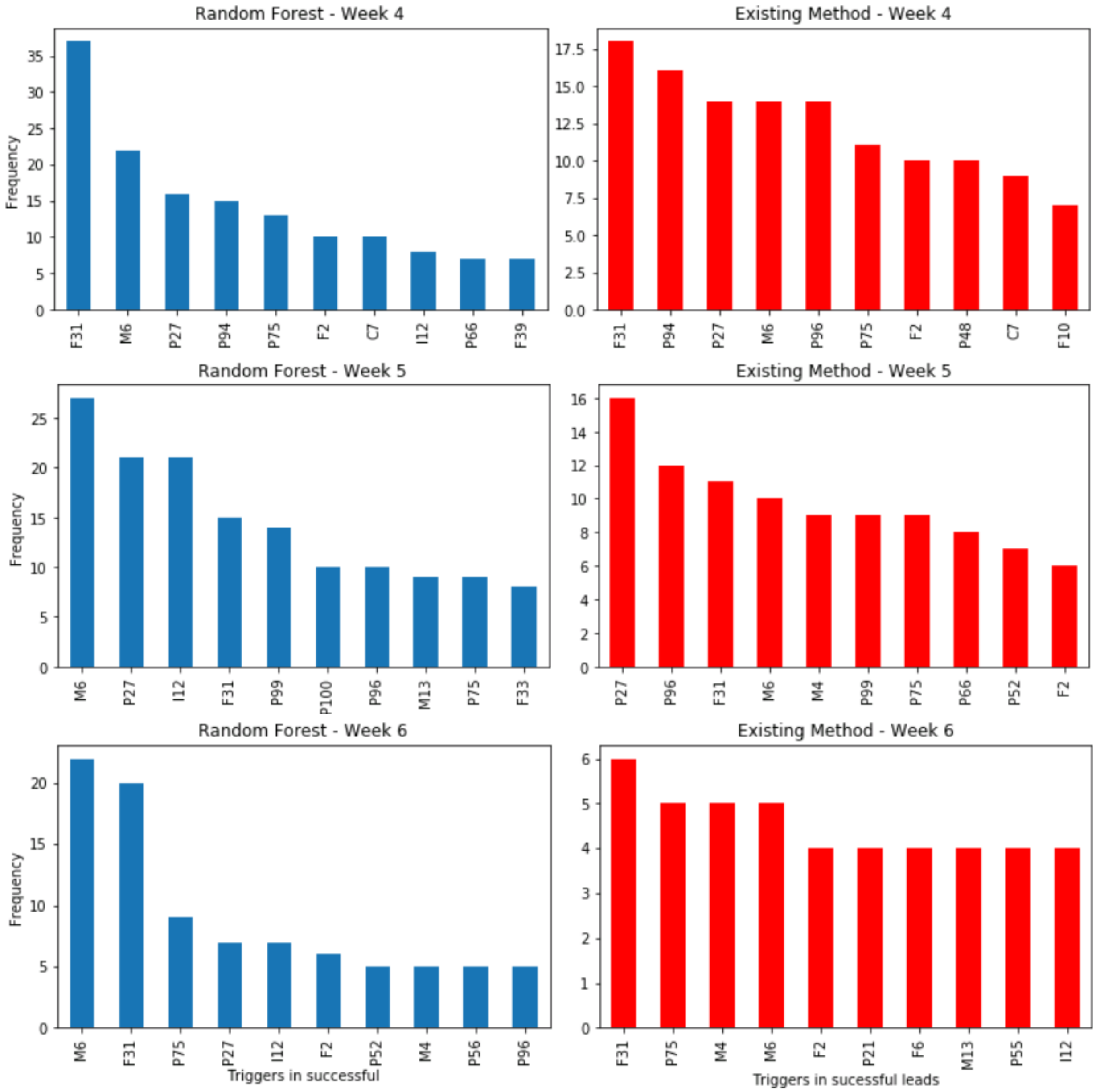


Figure 43: Top 10 most common triggers in successful leads per model over the 6 weeks

In the first 4 weeks, the random forest is overall more consistent with the success achieved, which is mainly driven by the M6 and F31 triggers. It can be seen that after re-training in week 4, there is a slight difference in the triggers for week 5 for the random forest. The new triggers P99, P100 and P96 are selected and proven to be successful. For the existing method, the doesn't seem to be a particular pattern, as the success is driven by more or less the same triggers over the 6 weeks, and it can be seen that even though the number of successful leads keeps decreasing, the method is unable to adapt due its relatively static scoring structure. The main point of the re-training in week 4 was to see if the model can use the information about the team ids as seen in section 5.2. In order to do so, 2 successful leads of the random forest model are selected from week 4 (before the change) and week 5 (after the change), and the contributions of features to the success probability are decomposed. The triggers making up each lead can be seen in Table 5 below:

Table 5: Triggers for each of the selected leads in weeks 4 & 5

1st Lead - Week 4	2nd Lead- Week 4	1st Lead- Week 5	2nd Lead- Week 5
F2	F2	M6	P73
F18	P66	P29	P27
F6	F31	F31	P96
F31	F40	C7	P99
P78	P78	team_id_5_00001	team_id_5_00001
team_id_5_00640	team_id_5_00550	-	-

Then, the top five highest feature contributions to the probability calculation for each lead can be looked at, as seen in Figure 44 below:

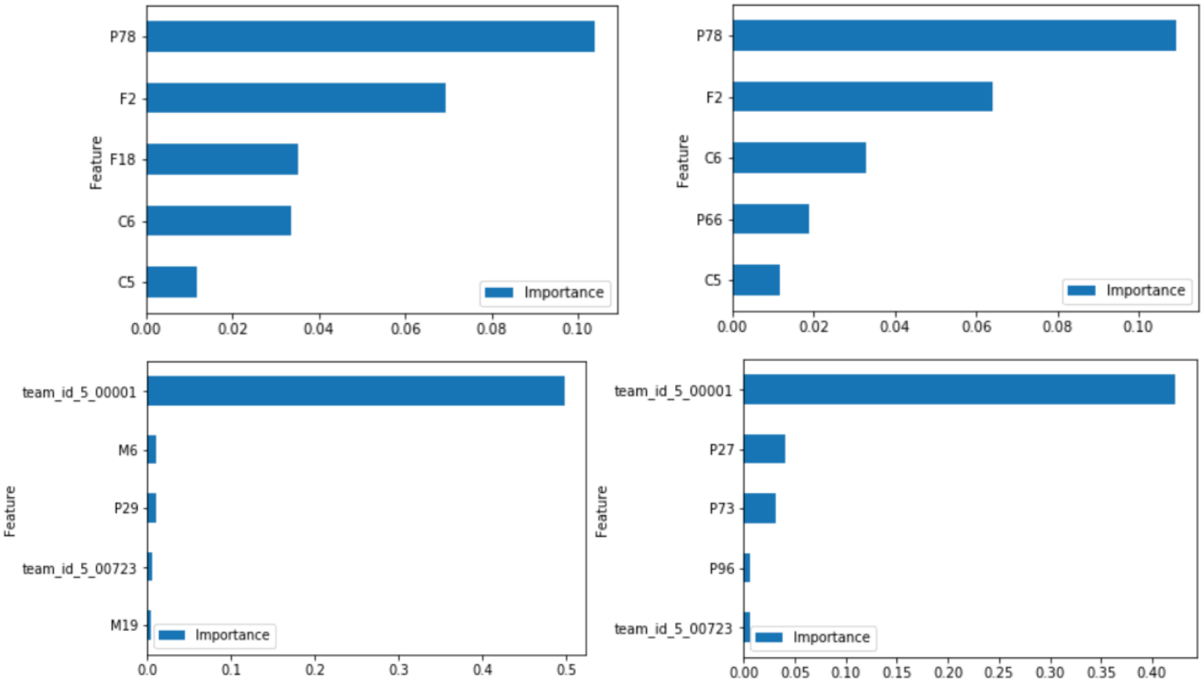


Figure 44: Feature importance of leads from week 4 (top row) and week 5 (bottom row)

It can be seen that after the changes of week 4, the most important factor in the decision making seems to be which team is receiving the lead. In addition to this, the amount of the contribution of the team id alone seems to be as high as the total contribution of other leads before the change was made. This point to increased amount of learning and less effect from the training data bias as compared to the leads of week 4. These figures suggest that the main reason why the random forest outperformed the existing method in weeks 4 and 5 is the addition of the team ids, and to a lesser extent the re-training with only the non-urgent leads.

Going back to decision interpretability, it can reasonably be said that the process of decision making is understood. The final decision is arrived at by aggregating the decisions made by a large number of decision trees, which as seen earlier in section 5.2 can have different decision paths and value different features. In theory, for a single decision every decision path over all trees can be looked at to give a full overview, but given that the forest is made of 2000 trees which have hundreds of nodes, this would not be feasible. This is why the decomposition of feature importance over the forest per decision was looked at earlier. This decomposition does give some clarification over which factors affected the decision being made the way it was, but to say it makes the decision fully interpretable is arguable. Overall, it can be said the workings of the model is closer to being white box than black box, but the exact interpretation of the decisions lies somewhere in the middle. In any case, the decision decomposition does add value to the decision making, as seen earlier when the difference between adding the team ids was observed or when based on the contribution of urgent triggers, it was decided to re-train the model only with non-urgent ones.

Furthermore, it is worth noting that the fact that there are a fixed minimum number of leads that must be sent out per week largely contributes to lowering the overall conversion rates of the random forest model. This is because a lot of leads where the probability of success is below 50% (that is to say the model has predicted that the outcome will be negative) will still be sent out to match the quota. In order to see this effect, the distribution of the probability of success calculated by the random forest model across successful and unsuccessful leads chosen by the model can be looked at, as seen in Figure 45 below:

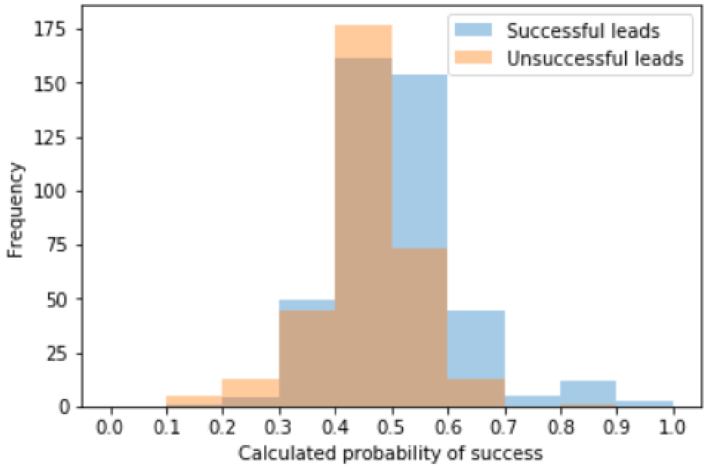


Figure 45: Calculated probability of success over the leads sent out by the random forest model

The overall distribution raises questions with regards to the capacity of the current lead generation system for achieving higher success. As calculated by the model, most of the leads seem to be less than 50% likely to be successful. It can be seen that the majority of the unsuccessful leads fall below the 50% threshold, whereas the majority of the successful are above this threshold.

Therefore, if the model was not forced to match a quota, many of the leads with below 50% probability of success would not have been sent out. Additionally, earlier in the results section it was explained that potentially there are too many leads being sent out due to the selection rule of sending out all urgent leads every week. While it's true that the non-urgent leads on average have higher conversion than urgent ones, the conversion is not so dominant (for example 80/ 90%) to justify this selection rule. As an alternative, the model could train and select all leads, urgent and non-urgent alike, and only send out the maximum number of leads that each team asked for instead. In order to see if this could have potentially improved the performance of the random forest model, the urgent leads have also been scored using the model, with the distribution of the probability of success across urgent and non-urgent leads over 6 weeks being seen in Figure 46 below:

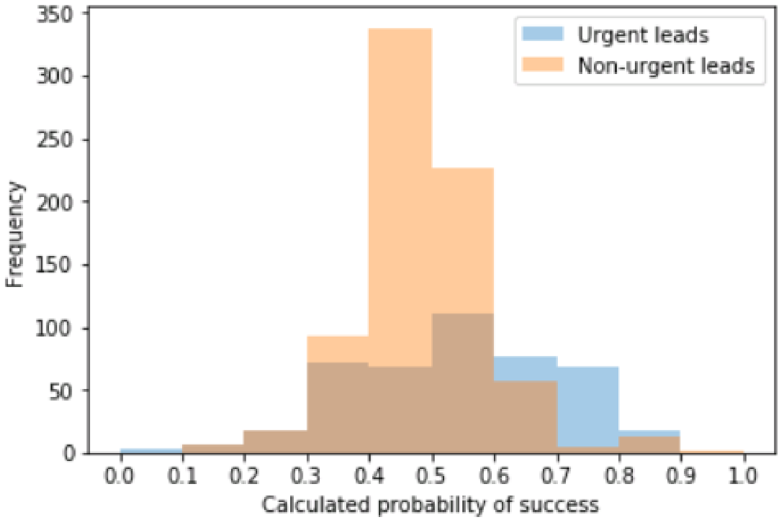


Figure 46: Calculated probability of success of urgent & non-urgent leads over 6 weeks

It can be seen that if this selection was made, the urgent leads would have gotten some form of priority in any case, as their calculated probability of success seems to be on average higher than non-urgent leads. This selection method would make the number of leads sent more reasonable, as well as improving the performance of the model by the fact that some of the low probability non-urgent leads which were forced to be sent out could have been replaced by high probability urgent ones.

Overall, with the addition of the team ids and the information learned from them, it can be seen that the random forest model does seem to be an improvement over the existing method. This can be seen both in terms of conversion and also the fact it is more adaptable with the feature selection remaining at a reasonably successful level given the dynamic nature of the data generation each week.

As discussed earlier, the input information to the model was restricted by the fact that the top five triggers were available per lead (and these five being selected by a similar scoring method as the one used for the leads) and that there was a considerable difference between the amount of leads sent and the amount that were labelled (for which usable feedback was received). Furthermore, in the testing the model is still limited by the quality of the pool of leads each week, as well as the selection method with regards to urgent leads which was utilized. Having presented and discussed the results of the testing, the survey results are presented next.

5.4 Banker Preference Survey Results

In this section the results of the survey sent out to the bankers are presented and discussed. The setup of the survey is as explained in section 3.4. The ten triggers which were sent out to the bankers can be seen in Table 6 below:

Table 6: The ten triggers selected for the survey

Trigger	Average Conversion	Urgent?
F48	0.588086	Yes
F26	0.599237	Yes
F39	0.619718	No
F18	0.593640	No
F2	0.650284	No
P75	0.626655	No
P66	0.632967	No
P21	0.603369	No
M6	0.527233	No
M4	0.615159	No

In total, teams from seven regions participated in the survey. To extract the preference, by counting the number of times each trigger is selected, a preference score between 0 to 9 is assigned to each trigger. The preferred triggers of the seven participating regions can be seen in Figure 47 below:

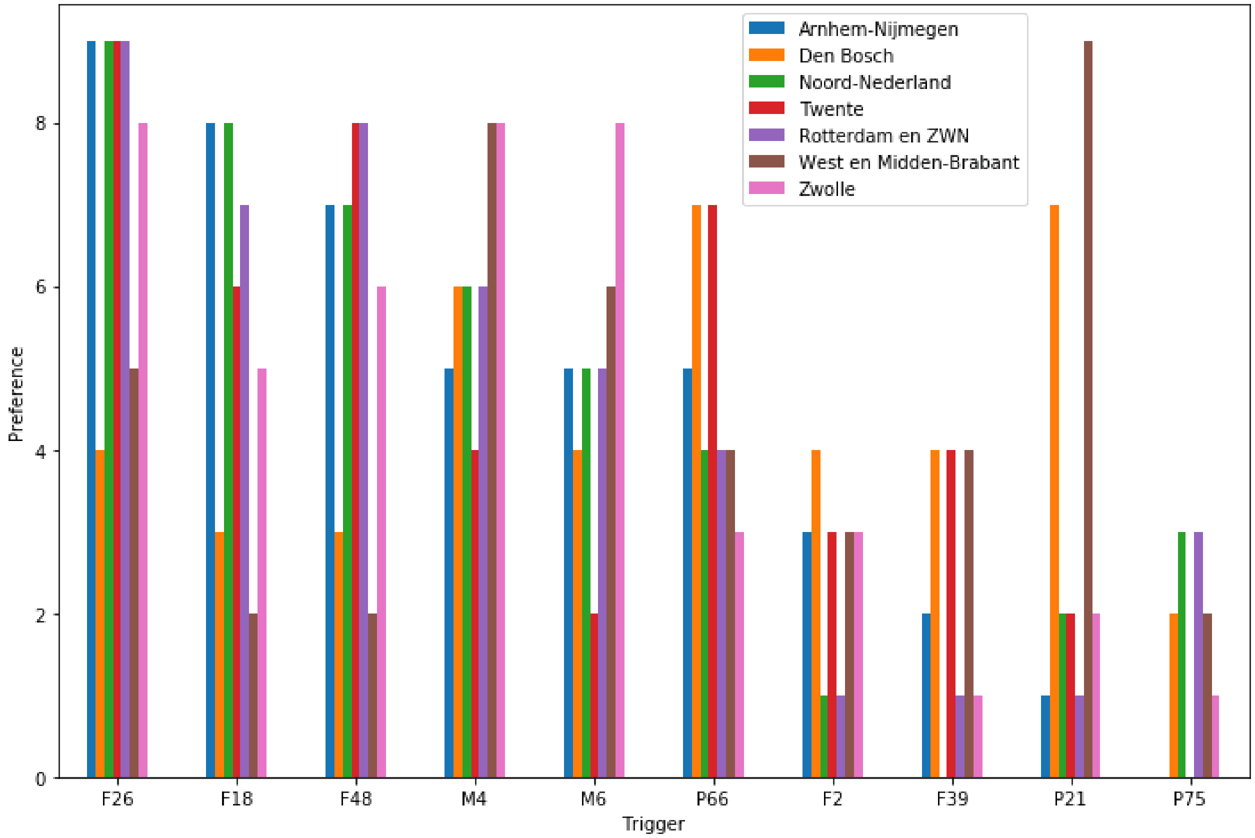
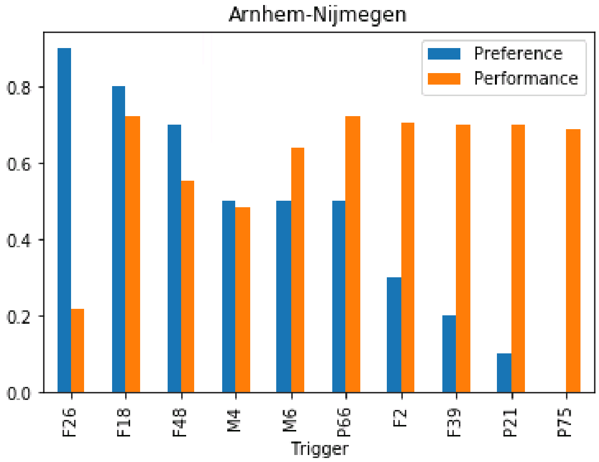


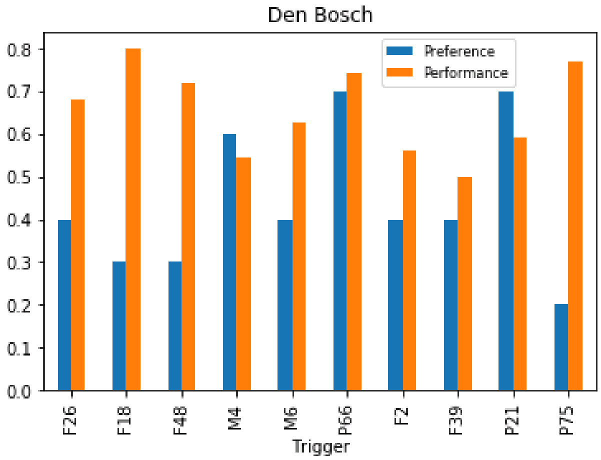
Figure 47: Trigger preferences of the seven participating regions

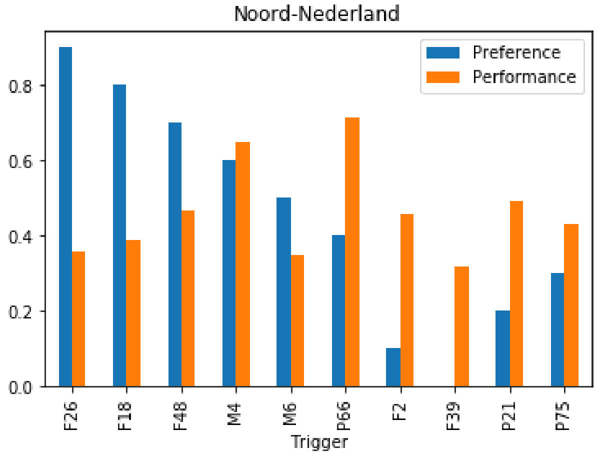
The most prominent fact from the figure is that the preferences of different regions are not particularly for the same triggers. It can be seen that the two urgent triggers F26 & F48 are generally popular, but not in all regions (most notably not in Den Bosch and West en Midden-Brabant). Additionally, certain regions like Noord-Nederland and Arnhem-Nijmegen seem to favor financial related (F) triggers more, whereas others like Den Bosch and Zwolle give equal preference to service (M) and product related (P) triggers.

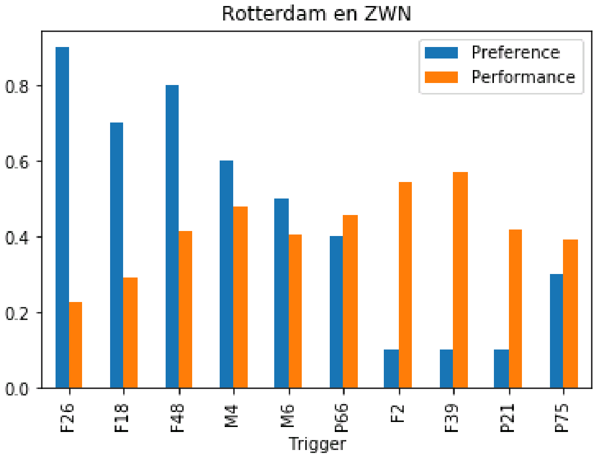
Interestingly, the two triggers P75 and F39 are among the least popular, despite both having an average conversion of over 60%. The reason behind this might be in the wording of the triggers. All of the other 8 triggers in the survey are clear indicators for a reason to call the customer. But P75 and F39 are speculative indicators (but based on logical observations), and they both start with the word ‘potentially’. This can be seen as a negative by the bankers, which could have a preference for more certain indicators.

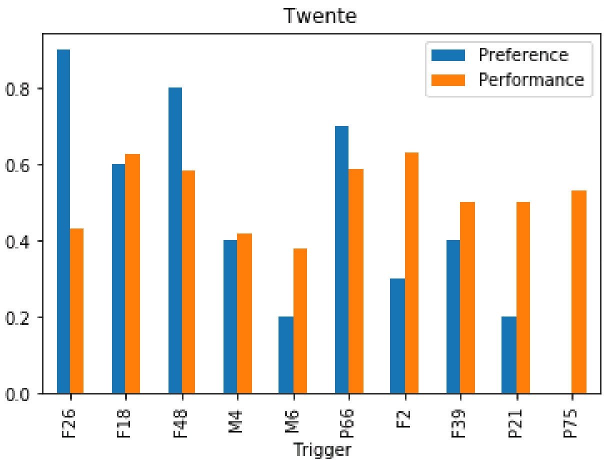
Having established the difference between preferences, it is then natural to look at whether this could potentially have any effect on performance. To do this, the preference score of each region, which has a value between 0 to 9, is divided by 10 (resulting in a value between 0 and 0.9). Then, this resulting value is plotted against the performance of the teams in the region on the specific trigger, in order to have a measure of preference against performance. The result can be seen in Figure 48 below for the seven regions:

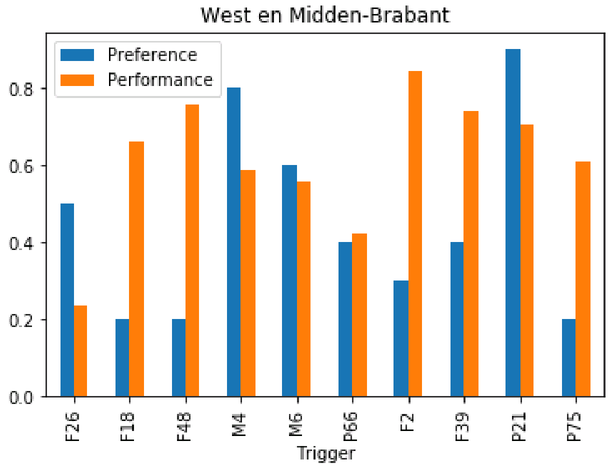

Arnhem-Nijmegen
Preference
Performance
0.0
0.2
0.4
0.6
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger


Den Bosch
Preference
Performance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger


Noord-Nederland
Preference
Performance
0.0
0.2
0.4
0.6
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger


Rotterdam en ZWN
Preference
Performance
0.0
0.2
0.4
0.6
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger


Twente
Preference
Performance
0.0
0.2
0.4
0.6
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger


West en Midden-Brabant
Preference
Performance
0.0
0.2
0.4
0.6
0.8
F26
F18
F48
M4
M6
P66
F2
F39
P21
P75
Trigger

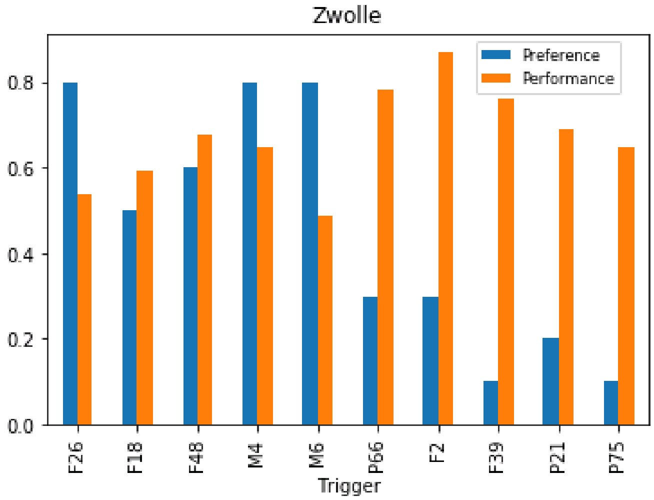


Figure 48: Performance vs preference of the participating regions over the ten triggers

If it was to be concluded that explicitly stated preference affects performance, the expectation would be to see triggers with higher preference having higher performance. Looking over the seven regions, this is simply not observed. While some regions like Den Bosch and West en Midden-Brabant have some level of correspondence between their preferences and performance, for most regions it seems the two are unrelated. In fact, for some regions like Zwolle or Arnhem-Nijmegen, it seems that they perform considerably better on their least preferred triggers. From a real-world perspective this cannot be justified, so it has to be attributed to randomness in the results, or the fact that the preferences are not stated rationally. It has to be also considered that there are 16 regions in the country, and only 7 agreed to participate in the survey. As a result, it is possible that these results are not representative of the sales teams as a whole.

As stated earlier, the survey results were not received in time to be processed for implementation in the scoring of leads. However, considering that the results only represent a subset of the regions and that even in that subset there seems to be no strong indication that the teams perform better on triggers which they say they prefer, there is no reason to suggest the results of this survey should in any way affect the scoring of the leads.

In any case, had the results been different, they could have been implemented in the scoring of the leads. In such a case, a method involving boosting based on the presence of preferred triggers (as a post processing operation) is proposed. That is to say, if a lead being sent to a team contains any of the top 5 preferred triggers for that team, the probability of success of the lead will boosted by a certain percentage per preferred trigger. Naturally, this may also be implemented in a reverse manner, with the presence of 5 least preferred triggers reducing the probability of success. Taking the simpler method by only considering favourable leads, the following boosting figures can be implemented:

Table 7: Example boosting figures for implementation of preferences into the scoring method

Preferred Trigger?	**Success Probability Boost (%)**
1st	5
2nd	4
3rd	3
4th	2
5th	1

Mathematically, that would be:

$$5.4.1 \quad Score(lead) = Max(1, Score(lead) + \sum_{x\epsilon 1}^{5} Boost(T_{p,x}))$$

Where Tp,x is the xth preferred trigger of the corresponding team. Then, any potential lead can be boosted by a maximum of 15%, so long as the probability does not exceed 1 (100%). Note that the probability of a team getting a lead of consisting of their top 5 preferred triggers is next to impossible, so in the case of such an implementation usually a boost in the range of 15% will not be observed.

Having presented and discussed the results and a potential method for implementation of preference into the scoring method, in the upcoming final chapter the project is concluded.

6 Overview & Conclusion

Having presented all the information in the previous chapters, in this chapter the research will be concluded. In the project overview, a summary of previous chapters is provided. Additionally, based on the results obtained, it will be discussed whether the priors stated in section 5.2 are supported or challenged. In the final section, the answers obtained based on this research for the SQs as defined in section 1.2 are presented and discussed. Furthermore, the main insights obtained from this research are highlighted, limitations are discussed and the potential for future research is investigated.

6.1 Project Overview

In this project, the technical goal was to investigate the potential integration of data analytics methods into the lead management system at the private banking department of ING and their usefulness in increasing conversion results. Additionally, it was aimed to achieve this technical goal while considering and ensuring appropriate measures for compliance with data protection laws such as the GDPR. Furthermore, it was aimed to investigate and introduce necessary measures to facilitate the governance of the algorithmic decision making process as part of GDPR compliance. In the first chapter, an introduction of the problem was provided by discussing the context of CRM as part of financial services, as well as the particular CRM in question in the case of this project. The limitations of the current system were addressed and the potential for modifications with regards to the scoring of the leads were proposed.

Following from that, in chapter 2, more theory was provided regarding CRM with a focus on the analytical component, where data analytics approaches have been used before in other areas of marketing. The case for the integration of a data analytics approach into the lead management system as opposed to the existing scoring method was made and based on theory about involvement of call center employees, a survey was suggested to collect data about the work preferences of private bankers in different regions. As well as explaining the relevance of the GDPR to this project in detail, the compliance measures undertaken by ING were examined, the most important of which is the balancing test. That is essentially a DPIA for all data processing activities within the company. An addition to this test was proposed based on decision interpretability, in order to help both the users of this system and the data agents.

In chapter 3, the technical methodology of the data analytics approach was presented and the theory of this methodology was briefly examined. Three machine learning algorithms were chosen based on potential accuracy, runtime and notional explainability to be tested for predictive analytics with the data. Furthermore, based on theory presented beforehand, the methodology of the survey with the bankers was presented.

In chapter 4, the execution of the methodology presented in the previous chapter was performed and presented. The potential for addition of more data to the algorithms was explored. Eventually, based on predictive accuracy, the random forest model was selected for testing against the existing scoring method.

In chapter 5, the concept of urgent and non-urgent leads was introduced. This was followed by a detailed investigation of the structure of the random forest model and the presentation of priors before the testing. Based on this investigation, it was proposed to divide the testing period into two phases. Then, the setting of the A/B testing with the operational integration of the data analytics methodology into the existing lead management infrastructure was presented. This was followed by the results of the A/B testing over both phases (6 weeks total). Furthermore, a discussion and evaluation of the results was presented, in order to gain more insight into how and why there were achieved. In the final section, the results of the survey were presented and discussed. Although in this project the survey results were not implemented in the lead scoring, a potential method for doing so was proposed.

Having presented an overview of the research, now a discussion of priors as defined in section 5.2 follows. Each prior can be seen below, followed by a discussion of the outcome based on the results obtained.

Prior 1: The urgent leads will have a higher conversion compared to non-urgent leads, as they contain some of the best performing triggers as seen in Figure 2 and some triggers preferred by the bankers.

Outcome: This was investigated in Figure 37 in section 5.3.2. It was clear that over the 6 weeks of testing, with the exception of one week, the urgent leads had on average about 10% higher conversion than non-urgent leads. Part of this is due to the make-up of the leads which include some of the highest performing triggers, and part of it can be attributed to the fact that the bankers may feel some form of pressure to do better with these leads, as they specifically stated a preference to receive more of them. Overall, this prior is supported by the results obtained.

Prior 2: Given the marketing-oriented nature of the private banking department, product (P) and financial (F) related triggers will be picked more frequently than others.

Outcome: Figure 42 in section 5.3.3 would be a good measure for this prior. There, it can be seen that indeed regardless of the scoring method, the two types of triggers most frequently selected are F and P triggers. For example, in week 1 this is 8 out of 13, in week 3 it is 9 out of 14 and in week 6 it is 10 out of 14 triggers. Therefore, this prior is supported by the results obtained.

Prior 3: After addition of team ids as features in week 4, in the random forest model the team ids will play an important role in the decision making and will increase the performance of the model.

Outcome: This can be examined in different ways. In Figure 39 in section 5.3.2, it can clearly be seen that after the addition of the team ids, the performance of the random forest model increases considerably and the model outperforms the existing scoring method. Furthermore, in Figure 41 in the same section can be seen that after the addition of the team ids, the model is consistently getting better at identifying successful leads in any probability threshold (even below 40%). Additionally, in Figure 44 in section 5.3.3 it can be seen that with the addition of the team ids, the overall learning of the model increases considerably and that the team ids are the major factor behind the decision making. Therefore, with reasonable confidence it can be concluded that the results obtained support this prior.

Prior 4: Based on the survey results, better performing regions have higher correspondence between their preferred trigger types and ones that they perform well on, which leads to a more motivated effort on the side of the bankers to have successful calls

Outcome: To investigate this prior, first Figure 18 from section 4.2.1 must be looked at. Based on that figure, it can be seen that out of the seven regions which participated in the survey, the highest performing ones are Arnhem-Nijmegen and Den Bosch, whereas the lowest performing ones are Noord-Nederland and Rotterdam en ZWN. Having this in mind, Figure 48 in section 5.4 can be looked at. There, it can be seen that there does not seem to be a strong correspondence between preference and performance for any particular region.

Even in some high performing regions (like Arnhem-Nijmegen), it seems that the triggers driving the performance are among the least preferred ones. This was also why it was concluded that at least in the case of this project, there doesn't seem to be much incentive to somehow include the explicitly stated preference of the call center agents in the scoring of the leads. Overall, it can be concluded that the results obtained challenge and to some extent contradict this prior.

Prior 5: Based on the survey results, the bankers will show a higher preference for urgent triggers than non-urgent, which would be consistent with the earlier agreement to always send all urgent leads.

Outcome: For this prior, Figure 47 in section 5.4 can be looked at. There, it can be seen the two urgent triggers F48 and F26 are first and third in terms of overall preference. This is in spite of the fact that they are in the lower half of the ten triggers sent to the bankers in terms of performance. This further insinuates that there is not much correlation between the performance of the triggers and the preferences of the bankers, and also raises questions with regards to whether the call center agents have adequate awareness of which types of leads usually drive their performance. Overall, the results obtained support this prior.

In the following final section, the SQs are answered, as well as a final discussion and potential future research recommendations being presented.

6.2 Conclusion & Future Steps

In the following parts, it is aimed to answer the SQs and arrive at a discussion of the main research question, based on all the information provided before this section. Below, each SQ can be seen, followed by an answer.

SQ1- What are the necessary measures for compliance to be ensured in the design of algorithms for data-driven lead management in light of the GDPR regulations?

Answer: There are many different clauses relevant to data processing, as explained in detail in chapter 2. The main compliance measure recommended for the GDPR is Data Privacy Impact Assessment (DPIA). In the case of ING, this measure is undertaken by the introduction of the balancing test, as well as limiting the usage of non-aggregated personal data (which in the case of this project, was not used in any extent). The balancing test considers five factors, relevant to the data processing operation: remoteness, context, nature of the data, possible consequences and safeguards.

As this project involved the usage of machine learning algorithms, a sixth factor was proposed, about the interpretability of decisions made by the algorithm. Based on the algorithm chosen (random forest), two measures were taken to interpret the decision making of the algorithm, both based on feature importance. The first was feature importance for the model as a whole, and the second was the decomposition of decision paths per decision in order to gain an insight about feature contribution to particular decisions. This was done in line with the sixth factor introduced earlier and in order to help facilitate the governance of the decision making process, by mitigating the risks of loss of accountability and human oversight associated with the use of such processes.

Based on the analysis of the decision making process, the decision making of the model was changed by retraining it, as well as adding team ids as features. After the addition, the decomposition showed that the team ids became the major decision making factor for the model. While it was argued that perhaps the decision making of the model is not fully interpretable (white box), it can be said that the decision making process is understood and to a reasonable extent the factors contributing to a decision are known as well.

Overall, the most important takeaway is to evaluate the application of data protection law like the GDPR to the project at hand, and to ensure the required compliance measures like the DPIA are created accordingly. Furthermore, based on the specifics of the project, the DPIA can be modified as necessary (example of this project being the addition of decision interpretability due to the use of machine learning algorithms).

SQ2- How can a data analytics driven scoring method be integrated into the existing lead management infrastructure at the bank?

Answer: While the existing infrastructure already had data storage in the form of a SQL database, the existing scoring method used only SQL queries as well. Therefore, for integrating a data analytics approach using Python, a combination of environments existing within the IT infrastructure of the bank had to be used. Due to the nature of the data, analytics operations could only be done on the secure cloud, where access to Python was also available. Therefore, Python scripts were used on the cloud to make a connection to the SQL database in order to import and process relevant data. Then, the output data had to be exported to Hive tables. The reason for this is that there are no direct writing privileges to the main SQL database. After exporting to Hive, with the right approvals from the relevant managers, the data could be exported back to the main SQL database by one of the IT teams with the correct writing privileges. From that point onwards, that data could be used in the UNICA environment for being sent out to the call centers.

Another challenge with the addition of the algorithm was the fact the features of the data were dynamic every week. In order to accommodate this, weekly batch learning was used. A re-index function was used to add the features that the new feature matrix was missing each week (giving them 0 value) and remove features that were not known to the model yet (by comparing column indices of the new feature matrix with that of the model). This way, the model learned in batch form every week, with the batch incrementally increasing in size per week based on the feedback of the previous week.

Overall, the most important takeaway is that depending on the project at hand, the possibilities for implementation of data analytics methods differ. The people in charge of the project should be flexible and ask for relevant support for integration of different environments for the purpose of successful implementation, if necessary (as was the case for this project).

SQ3- Based on a data driven approach, which n number of leads (n being the corresponding capacity for customer contact) have the highest probability of success (making a sale) at any particular time?

Answer: Based on the data analytics approach, the answer is it depends. The presence of certain triggers (like C1, C6 & C7) can highly contribute to the success of a lead, due to their critical nature, which puts pressure on both the bankers and the customers to accommodate a response. Furthermore, the model identified certain triggers like F31 and M6 as very important. This seems logical due to the nature of the triggers, which indicate the presence of high financial capital (F31) and the fact that a customer has not been contacted for a reasonably long time (M6).

Certainly the biggest takeaway comes from the decomposition of decision paths. It was noted that after the addition of the team ids, they were by far the most important factor in the decision making of the model. Therefore, the best answer to this question is it depends on who is receiving the leads. After that question is answered, then based on the teams different triggers can be important. This is not particularly surprising, as it was observed earlier in Figure 19 in section 4.2.1 that different teams can perform considerably differently on different types of triggers. This also suggests that perhaps when trying to increase success in a CRM involving call centers, companies should pay more attention to the call agents rather than putting all their focus on customer data and analytics. The combination of customer and call agent data could provide for more successful models, more content call agents and happier customers as a result of better service and communication.

SQ4- What is the effect of the preferences of the channels of contact (private bankers) on the success of the process and how can the lead distribution be optimized in this regard?

Answer: This was investigated in two ways. First, with the consultation of the bankers, 26 triggers which were deemed to be always good were classified as urgent and leads containing them were always sent out, regardless of scoring. Over the 6 weeks of testing, it was observed that on average these leads performed 10% better than the non-urgent leads. Therefore, in this sense, involving the bankers in the selection process proved to have a positive effect. Furthermore, a survey was conducted with seven participating regions in order to gain insight into the preferences of the bankers for certain triggers.

The results of the survey showed that at least in the seven participating regions, there does not seem to be a correspondence between the preferences and the performance of the bankers on different types of triggers. This however, can be project dependent. Therefore it is still worth taking a look at work preferences, and in previous studies it has been shown to be able to improve performance. As a result, an implementation method was proposed for incorporating explicit preferences into the scoring of leads. This can directly be applied to any project which uses probabilistic lead scoring, and may be adapted to be applied to other types of scoring.

Overall, there were many challenges in the different stages of this research. Some of these challenges were overcome (examples of which are explained above), and some could not be addressed within the scope of this project. It was shown earlier that while it is true that the urgent were more successful than non-urgent ones, this does not justify the selection of all urgent leads per week. It was argued that the data model also identified a priority for the urgent leads and could have selected a reasonable amount of them within the requested quota of each team per week. This way the total number of leads sent per week would also be decreased, which could increase the engagement of the bankers with the leads that they receive.

As explained earlier, there were two main information restrictions for the data model. The first one came from the fact that while only a maximum of five triggers were available per lead in the historical data, this does not represent all triggers that were available for those customers at the time the lead was created. This could not be addressed due to the fact that the total number of leads per customer per week were not saved, partially due to the very large amount of storage that would require to do so. However, saving this data could be valuable as it could give rise to the potential for starting the data modelling earlier than the lead stage.

Furthermore, the amount of leads that received feedback were only a fraction of the leads sent out, so if higher engagement was achieved in the previous year before the start of the project, the initial database for the training of the model could have been considerably larger. This, in combination with saving all triggers per customer could give rise to the possibility of using unsupervised machine learning methods for some form of clustering on the customer database, which could further narrow triggers per customer cluster. In Figure 49 below, the systematic overview of the proposed structure in chapter 2 can be seen, with the focus of this project and potential for future research highlighted:

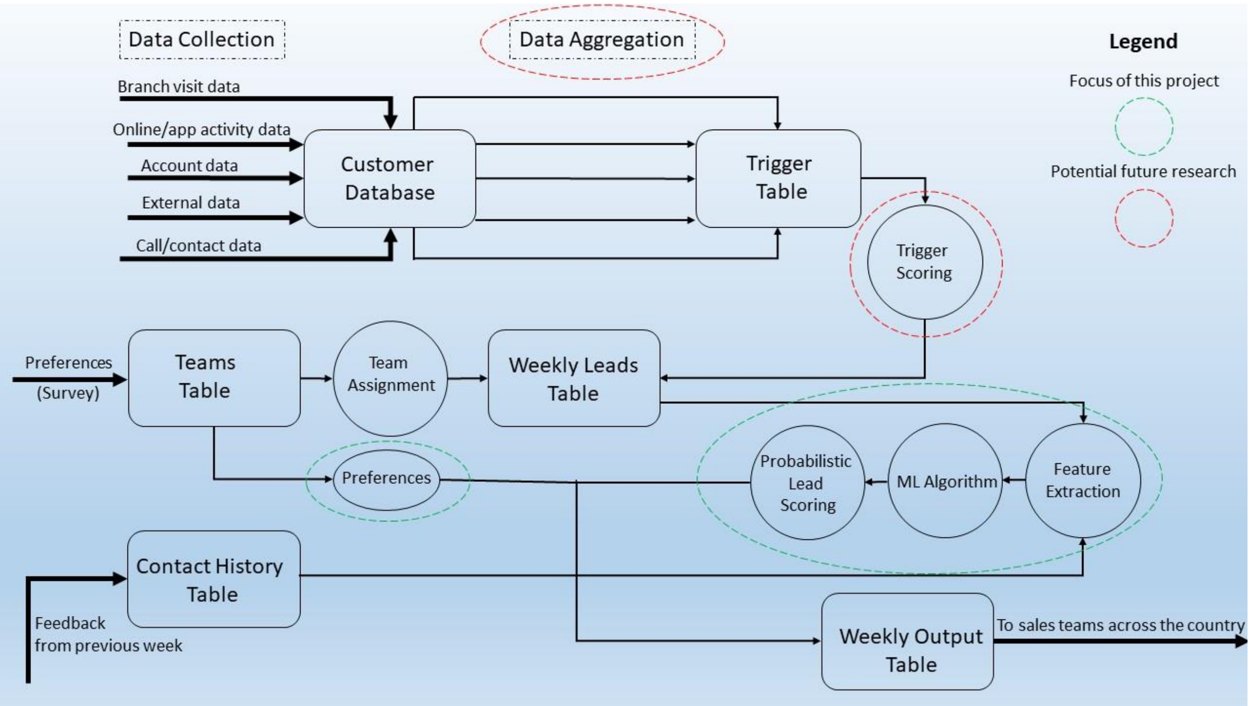


Figure 49: Potential areas for future research

As stated earlier, in this project the focus was on scoring the leads. If historical data about all triggers can be saved, the data modelling can start before the lead creation. Additionally, there are many triggers which have similar indications, but only differ in scale. For example, some of the F triggers indicate availability of investment capital at different monetary levels. These types of triggers can be aggregated to be one, and applied to different customer clusters (that is if customer clustering is performed as suggested earlier with unsupervised learning). Overall, future research can focus on the implementation of data modelling before the lead generation and employ smarter data aggregation techniques in the process of lead generation.

In summary, this project aimed to integrate data analytics methodology into the lead management system of the Private Banking department at ING, while ensuring compliance with data protection laws such as the GDPR. The measures in place were examined and it was proposed to add decision interpretability as part of the compliance measure in order to aid with the challenge of governance of the algorithmic decision making process. This later proved useful in improving the decision making of the model by modifying the feature set. Based on this, the inclusion of decision interpretability as a factor is recommended for any projects which use an algorithmic decision making process based on agents data, in order to mitigate the risks of loss of accountability and human oversight.

The technical results of the project have been reasonably successful, with good integration of the methodology into the existing infrastructure. In order for the integration to be achieved, several environments had to be used in combination as opposed to the existing method which could be performed only using one environment. A similar approach can be implemented in any lead management system, given that the appropriate IT infrastructure is present. The exact algorithm would depend on the case and data at hand, and must be selected through a process such as the testing explained in chapter 4 of this project. Additionally, successfully performing 6 weeks of live testing against the existing scoring method led to promising results with interesting insight about customer contact teams and triggers. Certain dominant triggers like F31 and M6 were identified, as well the fact that the most important factor in determining the success of any lead is the team which handles the lead. Based on this, it can be recommended any similar systems involving call center agents should also focus analytics operations on the call agents as well and not only on the customer side.

Furthermore, based on the results of a survey performed with participating regions, it was determined that the explicit preference of the bankers has no effect on the performance of the leads. This contradicted some previous research, and as a result it is recommended in similar projects the preferences of the call agents be examined. Furthermore, a method for the implementation of the results of such surveys into the lead selection was proposed which could be directly used with probabilistic lead scoring and adapted to be used with other forms of lead scoring. Based on the results and limitations, potential for future research was also presented, which could be pursued depending on the interest in further improving the lead management system and involving the data analytics and data science teams in potential collaborations. In the following final section, an informal personal reflection of the project and research experience as a whole is presented.

6.3 Personal Reflection

This section is written informally, as an overview of my experience and thoughts during and about the project and the whole research experience. The choice of performing the thesis externally was made with the idea of gaining industry experience and first-hand knowledge of applied data analytics and machine learning at a company, rather than from a purely academic perspective. Luckily, I did receive multiple offers for data-related projects for the thesis. The choice of ING was based on consultation with the chair of the project and due to the fact that the project was very well defined from the beginning, with the scope being fully determined.

The most obvious challenge in doing an external project was aligning the expectations from the university side and the company side. The company was solely interested in the main goal of the project, which was to improve the performance of the lead management system. The university asked for more of a scientific approach in explaining the background information, why the research is important and rigor in explaining the research methods. In the end, this is a scientific master’s thesis and as a result, this report was written to match the academic standards of the university.

It was interesting to see the process of applying machine learning techniques in a company to tackle a real world issue and create value for both the company and its customers. However, this application was not without its challenges. As we were working with aggregated customer data, no data was ever allowed to be taken off the secure cloud environment. This was one of the most important safeguards in place to ensure compliant processing of data. This was also the reason why a combination of different environments had to be used in order to integrate the machine learning algorithms as part of the lead management system. Furthermore, seeing how cookies work from the other side and their ability in data collection led to interesting discussions.

On the one hand, one might not be very comfortable with sharing their data from online activities using cookies. On the other hand, without this data, this type of research which in the end benefits the customers themselves as well would not be possible. The full debate of personal privacy versus business interests is out of the scope of this discussion. Nevertheless, it is important that regardless of any business interests, the customer is given some level of choice about how and for what purpose is their data is processed. Recently there have been encouraging developments in this regard, with all companies (ING included) having to provide opt-out options for customers who do not want their data to be used for any predictive analytics and personalized marketing activities. One has to wonder though, what would it take for this type of research to continue with full consent of the data agents which constitute its basis. In the end, the added value of any machine learning algorithm for a business is as good as the data on which the learning is based.

Personally, I believe there needs to be more transparency not only with regards to how data is collected (cookie consent, etc), but more so about why it's collected and how does this collection eventually benefit the customers themselves. Additionally, with increased usage of algorithms in more vital functions such as healthcare, human resources (HR) and crime, it is of the utmost importance that there is some level of interpretability about an algorithms decision making process. The usage of black-box algorithms on the basis of their supposedly better performance can hardly be justified when dealing with decisions which highly affect people's lives, as is the case in industries such as healthcare and HR. New approaches are being developed as more interpretable and explainable alternatives for popular algorithms. Interested readers are referred to the paper by Gunning (2017) for a good collection of such approaches. In the case of this project, the decomposition of the decision path proved to not only help understand the decision making of the random forest, but to also make informed decisions to modify the settings in order to improve the performance of the algorithm. This shows that understanding of machine learning algorithms is not only for the purpose being able to explain them, but can also help the users of the algorithm to improve performance, which is the main concern for supporters of more complex black-box methods. Furthermore, the challenge of governance can also be facilitated by the use of more interpretable methods. After all, what is not understood can hardly be governed. That is not to say using certain methods is the all-encompassing solution to the complex problem which is proper governance of AI and machine learning. Nevertheless, increasing understanding is a welcome first step.

Overall, during the course of this research, I learned a lot about the real world applications of machine learning algorithms to tackle a real world issue, as well as the challenges it brings with regards to safe processing and compliance with data protection laws and considering the challenge of governance of the algorithmic decision making process. While the experience of performing an external thesis was not easy one, it is one I highly recommend for any interested students who already have an idea about what type of projects they would like to do.

References

Abhishek, V., Fader, P., & Hosanagar, K. (2012). The Long Road to Online Conversion: A Model of Multi-Channel Attribution. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2158421

Bejtullahu-Michalopoulos, K., & Florin, M. (2018). The Governance of Decision-Making Algorithms.

Berman, R. (2013). Beyond the Last Touch: Attribution in Online Advertising. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2384211

Boulstridge, E., & Carrigan, M. (2000). Do consumers really care about corporate responsibility? Highlighting the attitude—behaviour gap. Journal of Communication Management, 4(4), 355–368. https://doi.org/10.1108/eb023532

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM). Business Process Management Journal, 9(5), 672–688. https://doi.org/10.1108/14637150310496758

Claesen, M., & De Moor, B. (2015). Hyperparameter Search in Machine Learning. Retrieved from http://arxiv.org/abs/1502.02127

Edwards, L., & Veale, M. (2017). Slave to the Algorithm? Why a Right to Explanation is Probably Not the Remedy You are Looking for. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2972855

European Banking Federation. (2018). Banking in Europe 2018: EBF Facts and Figures. Retrieved from https://www.ebf.eu/wp-content/uploads/2018/09/Banking-in-Europe-2018-EBF-Facts-and-Figures.pdf

Fayerman, M. (2002). Customer relationship management. New directions for institutional research, 2002 , 57-68. doi: https://doi.org/10.1002/ir.37

Flynn, R., Bellaby, P., & Ricci, M. (2009). The ‘Value-Action Gap’ in Public Attitudes towards Sustainable Energy: The Case of Hydrogen Energy. The Sociological Review, 57(2_suppl), 159–180. https://doi.org/10.1111/j.1467-954X.2010.01891.x

Genesys. (2009). The Cost of Poor Customer Service. Retrieved from http://www.ancoralearning.com.au/wp-content/uploads/2014/07/Genesys_Global_Survey09_screen.pdf

Gunning, D. (2017). Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency (DARPA). Accessed on 2017-09-09. http://www.darpa.mil/program/explainable-artificial-intelligence

Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computing, 28(3), 539–547. https://doi.org/10.1007/s11222-017-9746-6

Hsu, C.-W & Chang, C.-C & Lin, C.-J. (2003). A Practical Guide to Support Vector Classification. 101. 1396-1400.
ING. (2019). ING at a glance. Retrieved May 20, 2019, from https://www.ing.com/About-us/Profile/ING-at-a-glance.htm

Ji, W., Wang, X., & Zhang, D. (2016). A Probabilistic Multi-Touch Attribution Model for Online Advertising. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM 16. doi:10.1145/2983323.2983787

Kaggle. (2017). The State of ML and Data Science 2017 | Kaggle. Retrieved January 24, 2019, from https://www.kaggle.com/surveys/2017

Kennedy, T., Regehr, G., Rosenfield, J., Roberts, S. W., & Lingard, L. (2004). Exploring the gap between knowledge and behavior: a qualitative study of clinician action following an educational intervention. Academic Medicine : Journal of the Association of American Medical Colleges, 79(5), 386–393. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15107277

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, 1137--1143. Retrieved from https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529

Krogue, K. (2015). Top Inside Sales Challenges in 2015: New Research. Retrieved May 20, 2019, from https://www.forbes.com/sites/kenkrogue/2015/08/05/top-inside-sales-challenges-in-2015-new-research/#101fb21a44c7

Lin, Tom C. W. (16 April 2012). "A Behavioral Framework for Securities Risk". Seattle University Law Review. SSRN. SSRN 2040946.

Monat, J. P. (2011). Industrial sales lead conversion modeling. Marketing Intelligence & Planning, 29(2), 178–194. https://doi.org/10.1108/02634501111117610

Newell, S., & Marabelli, M. (2015). Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of ‘datification.’ The Journal of Strategic Information Systems, 24(1), 3–14. https://doi.org/10.1016/J.JSIS.2015.02.001

Parvatiyar, A., & Sheth, J. N. (2001). Customer Relationship Management: Emerging Practice, Process, and Discipline. Journal of Economic and Social Research (Vol. 3). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.2212&rep=rep1&type=pdf

Patel, R. (2015). Strategic Approach to Optimizing Leads Management Process – Capgemini Worldwide. Retrieved from https://www.capgemini.com/2015/10/strategic-approach-to-optimizing-leads-management-process/#

Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

Peng, C.-Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An Introduction to Logistic Regression Analysis and Reporting. The Journal of Educational Research, 96(1), 3–14. https://doi.org/10.1080/00220670209598786

PwC (2016). Trendrapport over fintech en nieuwe technologieen: De bancaire sector “ouderwets” innovatief. Retrieved from https://www.pwc.nl/nl/banken/assets/documents/pwc-Nvb-trendrapport-juli2016.pdf

Raileanu, L. E., & Stoffel, K. (2004). Theoretical Comparison between the Gini Index and Information Gain Criteria. Annals of Mathematics and Artificial Intelligence, 41(1), 77–93. https://doi.org/10.1023/B:AMAI.0000018580.96245.c6

Rutz, O. J., & Bucklin, R. E. (2011). From Generic to Branded: A Model of Spillover in Paid Search Advertising. Journal of Marketing Research, 48(1), 87–102. https://doi.org/10.1509/jmkr.48.1.87

Russel, S., & Norvig, P. (2009). The Theory of Learning (Third Ed.). *Artificial Intelligence: A Modern Approach* (pp 717-737). Upper Saddle River, New Jersey: Prentice Hall. Retrieved from https://dl.acm.org/citation.cfm?id=1671238

Saabas, A. (2014). Interpreting random forests | Diving into data. Retrieved June 18, 2019, from http://blog.datadive.net/interpreting-random-forests/

Sathyanarayana, S. (2014). A Gentle Introduction to Backpropagation. Numeric Insight, Inc Whitepaper.

Seni, G., & Elder, J. F. (2010). Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. Synthesis Lectures on Data Mining and Knowledge Discovery, 2(1), 1–126. https://doi.org/10.2200/S00240ED1V01Y200912DMK002

Shao, X., & Li, L. (2011). Data-driven multi-touch attribution models. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11 (p. 258). New York, New York, USA: ACM Press. https://doi.org/10.1145/2020408.2020453

Sheeran, P. (2002). Intention—Behavior Relations: A Conceptual and Empirical Review. European Review of Social Psychology, 12(1), 1–36. https://doi.org/10.1080/14792772143000003

Sisselman, M. E., & Whitt, W. (2009). Value-Based Routing and Preference-Based Routing in Customer Contact Centers. Production and Operations Management, 16(3), 277-291. doi:10.1111/j.1937-5956.2007.tb00259.x

Stine, B. (n.d.). Logistic Regression & Classification. Retrieved from http://www-stat.wharton.upenn.edu/~stine/mich/DM_05.pdf

Tavana, A.F., Fili, S., Tohidy, A., Vaghari, R., & Kakouie, S. (November 2013). "Theoretical Models of Customer Relationship Management in Organizations". International Journal of Business and Behavioral Sciences. 3 (11).

Teo, T., Devadoss, P., & Pan, S. (2006). Towards a holistic perspective of customer relationship management (crm) implementation: A case study of the housing and development board, singapore. Decision Support Systems, 42 (3), 1613 - 1627. doi: https://doi.org/10.1016/j.dss.2006.01.007

Theodorou, A., Wortham, R. H., & Bryson, J. J. (2017). Designing and implementing transparency for real time inspection of autonomous robots. Connection Science, 29(3), 230–241. https://doi.org/10.1080/09540091.2017.1310182

Tsiptsis, K. (2018). Data mining in the framework of analytical crm. Retrieved from https://www.slideshare.net/Tommy96/data-mining-in-the-framework-of-analytical-crm

Tweede Kamer der Staten-Generaal (2018). Antwoord op vragen van het lid Nijboer over het bericht ‘ABN Amro wil klantdata gebruiken voor advertenties.’ Retrieved from https://zoek.officielebekendmakingen.nl/ah-tk-20172018-2443.html

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Transparent, explainable, and accountable AI for robotics. Science Robotics, 2(6). https://doi.org/10.1126/scirobotics.aan6080

Walker, S. J. (2014). Big data: A revolution that will transform how we live, work, and think. International Journal of Advertising, 33 (1), 181-183. doi: 10.2501/IJA-33-1-181-183

Wolpert, D. H. (1996). The Lack of A Priori Distinctions Between Learning Algorithms. Neural Computation, 8(7), 1341–1390. https://doi.org/10.1162/neco.1996.8.7.1341

Yadagiri, M. M., Saini, S. K., & Sinha, R. (2015). A Non-parametric Approach to the Multi-channel Attribution Problem (pp. 338–352). Springer, Cham. https://doi.org/10.1007/978-3-319-26190-4_23

Y. Ng, A., & Jordan, M. (2002). On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. Adv. Neural Inf. Process. Sys. 2.

Zheng, A. (2015). Evaluating Machine Learning Models. Retrieved June 18, 2019, from https://www.oreilly.com/ideas/evaluating-machine-learning-models/page/5/hyperparameter-tuning

Appendix I: Additional Tables

Table A-1: Accuracy score & runtime of algorithms per fold in the initial setting

model_name	fold	accuracy	runtime (seconds)
LogisticRegression	0	0.633	2.006
LogisticRegression	1	0.624	2.006
LogisticRegression	2	0.628	2.006
LogisticRegression	3	0.637	2.006
LogisticRegression	4	0.630	2.006
RandomForestClassifier	0	0.597	13.307
RandomForestClassifier	1	0.586	13.307
RandomForestClassifier	2	0.595	13.307
RandomForestClassifier	3	0.598	13.307
RandomForestClassifier	4	0.600	13.307
MLPClassifier	0	0.587	230.381
MLPClassifier	1	0.589	230.381
MLPClassifier	2	0.597	230.381
MLPClassifier	3	0.589	230.381
MLPClassifier	4	0.592	230.381

Table A-2: Accuracy score & runtime of algorithms per fold with the compact feature matrix

model_name	fold	accuracy	runtime (seconds)
LogisticRegression	0	0.643	0.693
LogisticRegression	1	0.635	0.693
LogisticRegression	2	0.640	0.693
LogisticRegression	3	0.650	0.693
LogisticRegression	4	0.633	0.693
RandomForestClassifier	0	0.604	2.653
RandomForestClassifier	1	0.603	2.653
RandomForestClassifier	2	0.598	2.653
RandomForestClassifier	3	0.595	2.653
RandomForestClassifier	4	0.601	2.653
MLPClassifier	0	0.602	174.562
MLPClassifier	1	0.590	174.562
MLPClassifier	2	0.603	174.562
MLPClassifier	3	0.586	174.562
MLPClassifier	4	0.598	174.562

Table A-3: Accuracy score & runtime of algorithms per fold after the addition of team ids

model_name	fold	accuracy	runtime (seconds)
LogisticRegression	0	0.707	0.974
LogisticRegression	1	0.702	0.974
LogisticRegression	2	0.703	0.974
LogisticRegression	3	0.699	0.974
LogisticRegression	4	0.703	0.974
RandomForestClassifier	0	0.661	5.454
RandomForestClassifier	1	0.672	5.454
RandomForestClassifier	2	0.678	5.454
RandomForestClassifier	3	0.671	5.454
RandomForestClassifier	4	0.663	5.454
MLPClassifier	0	0.665	203.791
MLPClassifier	1	0.655	203.791
MLPClassifier	2	0.668	203.791
MLPClassifier	3	0.660	203.791
MLPClassifier	4	0.662	203.791

Table A-4: Accuracy score & runtime of algorithms per fold after the addition of customer potential

model_name	fold	accuracy	runtime (seconds)
LogisticRegression	0	0.714	1.153
LogisticRegression	1	0.714	1.153
LogisticRegression	2	0.718	1.153
LogisticRegression	3	0.718	1.153
LogisticRegression	4	0.716	1.153
RandomForestClassifier	0	0.686	5.340
RandomForestClassifier	1	0.687	5.340
RandomForestClassifier	2	0.694	5.340
RandomForestClassifier	3	0.687	5.340
RandomForestClassifier	4	0.691	5.340
MLPClassifier	0	0.682	220.650
MLPClassifier	1	0.677	220.650
MLPClassifier	2	0.679	220.650
MLPClassifier	3	0.678	220.650
MLPClassifier	4	0.677	220.650

Table A-5: Hyperparameters per algorithm

Algorithm	Hyperparameter	Meaning	Default Value	Optimized Value
Logistic Regression	solver	Loss function minimization method	‘liblinear’	‘newton-cg’
Logistic Regression	C	Penalty term to avoid overfitting	1	1
Random Forest	n_estimators	Number of decision trees to aggregate from	10	2000
Random Forest	max_depth	Maximum level of depth reached before stopping	None	120
Random Forest	max_features	Number of features to consider at each split	‘auto’	‘sqrt’
Random Forest	min_samples_split	Minimum number of samples required to split a node	2	2
Random Forest	min_samples_leaf	Minimum number of samples required at any leaf node	1	3
Random Forest	bootstrap	Whether to use random sampling to build each tree (True) or use the whole data to build each tree (False)	False	True
MLP Classifier	hidden_layers_sizes	The number of hidden layers and their size	(100,)	(50, 50, 50)
MLP Classifier	activation	The activation function in the hidden layers	‘relu’	‘tanh’
MLP Classifier	alpha	Penalty term to avoid overfitting	0.0001	0.16
MLP Classifier	solver	Loss function minimization method	‘adam’	‘adam’
MLP Classifier	learning_rate	The learning rate for the minimization of the loss function	‘adaptive’’	‘adaptive’

Table A-6: Accuracy score & runtime of algorithms per fold after hyperparameter tuning

model_name	fold	accuracy	total runtime (minutes)
LogisticRegression	0	0.706	1.4
LogisticRegression	1	0.695	1.4
LogisticRegression	2	0.712	1.4
LogisticRegression	3	0.693	1.4
LogisticRegression	4	0.699	1.4
RandomForestClassifier	0	0.709	558.9
RandomForestClassifier	1	0.709	558.9
RandomForestClassifier	2	0.714	558.9
RandomForestClassifier	3	0.697	558.9
RandomForestClassifier	4	0.701	558.9
MLPClassifier	0	0.704	473.6
MLPClassifier	1	0.690	473.6
MLPClassifier	2	0.718	473.6
MLPClassifier	3	0.686	473.6
MLPClassifier	4	0.701	473.6

Table A-7: List of all urgent triggers

Count	**Trigger**
1	F48
2	F25
3	P49
4	C1
6	M20
8	F35
9	F15
10	F8
11	P69
12	C5
13	P40
14	F35
15	F26
17	F5
18	M10
19	P7
20	P10
21	P5
22	M9
23	F22
24	P72
25	O3
26	C6